

26.101/H/06



PENGATURAN PERGERAKAN ROBOT UNTUK APLIKASI PENGEBORAN PCB

Oleh :

Adam Ridiantho Muhamad
NRP. 2203 204 002

RTE
629.892
Muh
p-1
2005



PERPUSTAKAAN ITS	
Tgl. Terima	10 - 2 - 06
Terima Dari	H
No. Agenda Prp.	224713

**PROGRAM STUDI MAGISTER
BIDANG KEAHLIAN ELEKTRONIKA
JURUSAN TEKNIK ELEKTRO
FAKULTAS TEKNOLOGI INDUSTRI
INSTITUT TEKNOLOGI SEPULUH NOPEMBER
SURABAYA
2005**

PENGATURAN PERGERAKAN ROBOT UNTUK APLIKASI PENGEBORAN PCB

**Tesis disusun untuk memenuhi salah satu syarat memperoleh gelar
Magister Teknik (MT)
di
Institut Teknologi Sepuluh Nopember**

Oleh :

**Adam Ridiantho Muhamad
Nrp. 2203 204 002**

**Tanggal Ujian : 22 Juli 2005
Periode Wisuda : September 2005**

Disetujui oleh Tim Penguji Tesis :


**1. Ir. Djoko Purwanto, M.Eng., Ph.D
NIP. 131 879 397**

(Pembimbing I)


**2. Ir. Hendra Kusuma, M.Eng
NIP. 131 846 104**

(Pembimbing II)


**3. Prof. Dr. Ir. Mauridhi Hery P., M.Eng.
NIP. 131 569 364**



**6. Ir. Harris Pirngadi, MT
NIP. 131 843 903**

(Penguji)


**4. Prof. Dr. Siswandono, M.S., Apt.
NIP. 130 809 079**


**7. Rachmad Setiawan, ST. MT
NIP. 132 134 651**

(Penguji)


**5. Ir. Dadet Pramadihanto, M.Eng., Ph.D
NIP. 131 803 696**


**8. Ir. Totok Mujiono, M.L.Kom
NIP. 131 846 106**

(Penguji)

Direktur Program Pascasarjana,


**Prof. Ir. Happy Ratna S., MSc., Ph.D
NIP. 130 541 829**



PENGATURAN PERGERAKAN ROBOT UNTUK APLIKASI PENGEBORAN PCB

Nama mahasiswa : Adam Ridiantho Muhamad
NRP : 2203 204 002
Pembimbing : Ir. Djoko Purwanto, M.Eng. Ph.D.
Co-Pembimbing : Ir. Hendra Kusuma, M.Eng.

Abstrak

Pembuatan PCB merupakan suatu kegiatan yang biasa dilakukan oleh mahasiswa program studi elektronika ataupun industri yang bergerak dibidang elektronika. Demikian pula penggunaan peralatan penunjang untuk pembuatan PCB. Salah satu alat dalam proses pembuatan PCB adalah mesin bor. Mesin bor yang digunakan mulai dari yang manual sampai yang otomatis (*robot*). Penelitian pun diarahkan untuk menciptakan mesin bor otomatis (*robot*) yang mampu mengebor dengan cepat dan efisien tanpa ada satu lubang yang terlewatkan.

Dalam tesis ini akan dibahas metoda pengeboran PCB yang dilakukan oleh robot dengan kemampuan pemetaan posisi lubang dan perencanaan lintasan (*path planning*) secara otomatis. Pemetaan pada umumnya dibentuk secara manual dengan cara memasukkan posisi koordinat setiap lubang (*hole*), pemetaan posisi secara manual biasanya berurutan berdasarkan kesamaan koordinat x atau koordinat y untuk setiap lubang. Dengan metoda manual seperti ini lintasan gerak robot semakin jauh / panjang dan tidak efisien yang mengakibatkan proses pengeboran PCB membutuhkan waktu yang lama. Dalam tesis ini akan ditunjukkan penggunaan metoda genetic algoritma yang dilakukan robot sebagai pemetaan posisi. Perencanaan lintasan dilakukan oleh robot agar semua posisi lubang dalam PCB dapat dibor. Optimasi perencanaan lintasan dibutuhkan agar dihasilkan jarak lintasan yang terpendek berkaitan dengan waktu pengeboran yang singkat. Efisiensi jarak tempuh dengan menggunakan metoda genetik algoritma tergantung dari total posisi lubang. Pada total posisi lubang yang sedikit metoda genetik algoritma tidak efisien, pada total lubang efisiensi metoda genetik algoritma mencapai 25 % dan total lubang yang banyak efisiensi mencapai lebih dari 50 %.

Penelitian ini diharapkan memberi kontribusi pada pengembangan teknologi *robot* pengeboran PCB.

Kata kunci : robotika, pemetaan, perencanaan lintasan.

ROBOT MOVEMENT CONTROLLING FOR PCB DRILLING

Name : Adam Ridiantho Muhamad
NRP : 2203 204 002
Supervisor : Ir. Djoko Purwanto, M.Eng. PhD
Sub-supervisor : Ir. Hendra Kusuma, M.Eng

Abstract

PCB making is one of activity of student at electronics engineering or electronics industry. There are many supporting tools in PCB making. One of supporting tool in PCB making is drill machine. Drill machine that be used from manual to automatic (robot). Research is to be aimed to create automatic drill machine (robot) that can drill fast and efficient without any holes pass.

This thesis will study PCB drilling method that executed by robot with ability of automatically holes position mapping and path planning. Generally, mapping is formed manually by entering coordinate of each hole that usually in series with uniformity of x or y coordinate. With this method path of robot movement become more distant and inefficient that will need more time to complete the task. This thesis will show the use of genetic algorithm method in robot's position mapping. The robot will do planning path in order to drill every hole at the right position. Planning path optimization is needed to get the shortest path in order to reduce time needed to drill. The efficiency of genetic algorithm method is depending on the amount of holes. For PCB with few holes, genetic algorithm is inefficient. For PCB with more holes the efficiency is attaining 25%. For PCB with many holes, the efficiency is more than 50%.

This research is to be aimed to give contribution to development of PCB drilling robot technology.

Keywords : robotics, mapping, path planning

KATA PENGANTAR

Puji dan syukur penulis panjatkan kepada Allah SWT yang Maha Pengasih dan Penyayang atas berkat dan karunia yang diberikan sehingga penulisan laporan tesis ini dapat terselesaikan. Laporan tesis ini disusun dengan judul :

“ PENGATURAN PERGERAKAN ROBOT UNTUK APLIKASI PENGEBORAN PCB ”

Segala upaya dan kemampuan telah penulis lakukan demi kesempurnaan laporan ini sebagai pertanggung jawaban ilmiah. Penulis menyadari laporan ini tidak akan dapat tersusun tanpa bantuan dan bimbingan dari berbagai pihak. Untuk itu penulis menghaturkan rasa terima kasih dari lubuk hati yang dalam kepada yang terhormat:

- Bapak Ir. Djoko Purwanto , M.Eng, Ph.D. selaku koordinator program pasca sarjana Elektronika ITS merangkap sebagai dosen pembimbing.
- Bapak Ir. Hendra Kusuma, M.Eng. selaku dosen wali merangkap dosen pembimbing.
- Kawan-kawan seangkatan Program Pasca Sarjan Elektronika-ITS 2004; Pujiono, Morlan Pardede, Kristiono, Zaenal Arif, Indrasno, Nurkholis, M.Basyir, Mardi, Catur dan Endah sebagai teman diskusi selama studi.
- Orang tua tercinta, adik tercinta dan keponakan tersayang : Salsa dan Gaza yang telah banyak memberi dorongan dan doa serta pengorbanan demi kelangsungan studi.

- Semua pihak yang tidak dapat kami sebutkan satu persatu yang dengan keikhlasannya telah banyak membantu kelancaran pelaksanaan penelitian hingga penulisan laporan Tesis ini.

Penulis menyadari bahwa laporan tesis ini jauh dari sempurna mengingat keterbatasan yang penulis miliki. Untuk itu kritik dan saran yang bersifat membangun demi kesempurnaan tesis ini sangat penulis harapkan.

Akhirnya penulis berharap semoga tesis ini bermanfaat bagi perkembangan ilmu elektronika khususnya robotika di jurusan Teknik Elektro-ITS Surabaya dan menambah wawasan bagi yang membacanya.

Surabaya, Juli 2005

Penulis

DAFTAR ISI

	Hal
Halaman Judul	i
Lembar pengesahan	ii
Abstrak	iii
Abstract	iv
KATA PENGANTAR	v
DAFTAR ISI	vii
DAFTAR GAMBAR	x
DAFTAR TABEL	xiii
 BAB 1 PENDAHULUAN	
1.1 Latar Belakang	1
1.2 Perumusan Masalah	1
1.3 Tinjauan Penelitian	3
1.4 Manfaat Penelitian	3
 BAB II DASAR TEORI	
2.1 MOTOR DC	4
2.1.1 Prinsip Kerja Motor DC	5
2.1.2 Konstruksi Motor DC	6
2.2 Kontroller Proporsional Integral	6
2.2.1 Kontroller PI	7
2.3 Jaringan Syaraf (Neural Network)	8
2.3.1 Pendahuluan	8
2.3.2 Model Jaringan Syaraf	9

2.3.2.1 Jaringan Umpan Maju (Feedforward Network).....	12
2.3.2.2 Jaringan Umpan Balik (Feedback Network)	14
2.3.3 Algoritma Backpropagation	17
2.3.3.1 Definisi	18
2.3.3.2 Gradient	18
2.3.3.3 Aktivitas Maju (Forward Activation)	19
2.3.3.4 Menghitung Error Keluaran (Calculation Output Error)	19
2.3.3.5 Error Backpropagation	20
2.4 Pembelajaran Neural Network Based Tuner (NNPT)	21
2.5 Metoda Genetik Algoritma	23
2.5.1 Definisi Penting Dalam Algoritma Genetika	23
2.5.2 Langkah – Langkah Yang Harus Dilakukan Dalam Menggunakan Algoritma Genetika	24
2.5.3 Algoritma Genetika Dalam Perangkat Lunak	24
BAB III PERANCANGAN SISTEM	
3.1 Konstruksi Mekanik	27
3.1.1 Konstruksi Mekanik Sumbu X	27
3.1.2 Konstruksi Mekanik Sumbu Y	28
3.1.3 Konstruksi Mekanik Sumbu Z	29
3.2 Konstruksi Elektronik	29
3.2.1 Pengaturan Kecepatan Motor DC	29
3.2.2 Sensor Kecepatan	33
3.2.3 Sensor Jarak	36
3.3 Konstruksi Software	39

3.3.1 Metoda Neural Network Pi Controller	39
3.3.2 Metoda Konversi File PCB	42
3.3.3 Metoda Genetik Algoritma	45

BAB IV PENGUJIAN SISTEM

4.1 Pengujian Sensor Kecepatan	48
4.2 Pengujian Ladder R-2r	51
4.3 Pengujian Sensor Posisi	52
4.3.1 Sumbu X	52
4.3.2 Sumbu Y	54
4.4. Pengujian NNPT	56
4.5 Pengujian Algoritma Genetika	61

BAB V KESIMPULAN DAN SARAN

5.1 Kesimpulan	68
5.2 Saran	70

DAFTAR PUSTAKA	71
-----------------------------	-----------

LAMPIRAN

Rangkaian Motor Stepper	72
Rangkaian Ladder R-2r & Sensor Kecepatan	73
Rangkaian 89c51 Dan Sensor	74

DAFTAR GAMBAR

Gambar 2.1. Diagram blok konstruksi elektronik	4
Gambar 2.2. Prinsip kerja Motor DC.....	5
Gambar 2.3. Konstruksi Motor DC.....	6
Gambar 2.4. Diagram blok PI	7
Gambar 2.5. Jaringan Lapis Jamak (Multi-layer Network)	11
Gambar 2.6. Jaringan Umpan Maju Single-Layer	14
Gambar 2.7. Jaringan Umpan Balik Waktu Diskrit.....	16
Gambar 2.8. Jaringan Syaraf Sederhana	17
Gambar 2.9. Diagram Blok Pengaturan Motor DC dengan PI Kontroler dan JST.....	21
Gambar 2.10. Diagram blok Algoritma genetika	24
Gambar 2.11. Mesin Roulette	25
Gambar 2.12. Diagram blok Crossover	25
Gambar 2.13. Diagram blok Mutasi Gen.....	26
Gambar 3.1. Konstruksi Mekanik sumbu X	27
Gambar 3.2. Konstruksi Mekanik sumbu Y	28
Gambar 3.3. Konstruksi Mekanik sumbu Z.....	29
Gambar 3.4. Ladder R-2R.....	30
Gambar 3.5. Langkah 1.a	31
Gambar 3.6. Langkah 1.b.....	31
Gambar 3.7. Langkah 1.c.....	31
Gambar 3.8. Langkah 2.....	32
Gambar 3.9. Langkah 3.....	32

Gambar 3.10. Langkah 4.....	33
Gambar 3.11. Non inverting amplifier.....	33
Gambar 3.12. Rangkaian lampu 220 Vac.....	34
Gambar 3.13. Rangkaian Sederhana.....	34
Gambar 3.14. Cara kerja motor DC.....	35
Gambar 3.15. Rangkaian comparator.....	36
Gambar 3.16. Rangkaian optocoupler.....	38
Gambar 3.17. Arsitektur Backpro.....	39
Gambar 3.18. Menu snap grid.....	43
Gambar 3.19. Menu Set origin.....	43
Gambar 3.20. Algoritma Genetika.....	45
Gambar 4.1 Time/ Dive 0.5 uS, Data Ladder R-2R = 32H.....	48
Gambar 4.2. Time/ Dive 0.5 uS, Data Ladder R-2R = 64H.....	48
Gambar 4.3. Time/ Dive 0.5 uS, Data Ladder R-2R = 96H.....	49
Gambar 4.4. Time/ Dive 0.5 uS, Data Ladder R-2R = 0C8H.....	49
Gambar 4.5. Time/ Dive 0.1 uS, Data Ladder R-2R = 032H.....	50
Gambar 4.6. Time/ Dive 0.1 uS, Data Ladder R-2R = 64H.....	50
Gambar 4.7. Time/ Dive 0.1 uS, Data Ladder R-2R = 96H.....	50
Gambar 4.8. Time/ Dive 0.1 uS, Data Ladder R-2R = 0C8H.....	51
Gambar 4.9. Percobaan 1 dgn setpoint 1000 Rpm	56
Gambar 4.10. Percobaan 2 dgn setpoint 1000 Rpm	56
Gambar 4.11.. Percobaan 1 dgn setpoint 5000 Rpm	57
Gambar 4.12. Percobaan 1 dgn setpoint 10.000 Rpm	57
Gambar 4.13. Percobaan 2 dgn setpoint 10.000 Rpm	57

Gambar 4.14. Percobaan 1 dgn setpoint 15.000 Rpm	58
Gambar 4.15. Percobaan 2 dgn setpoint 15.000 Rpm	58
Gambar 4.16. Percobaan 1 dgn setpoint 20.000 Rpm	58
Gambar 4.17. Percobaan 2 dgn setpoint 20.000 Rpm	59
Gambar 4.18. Percobaan 1 dgn setpoint 25.000 Rpm	59
Gambar 4.19. Percobaan 2 dgn setpoint 25.000 Rpm	59
Gambar 4.20. Percobaan 1 dgn setpoint 30.000 Rpm	60
Gambar 4.21. Percobaan 2 dgn setpoint 30.000 Rpm	60
Gambar 4.22. Percobaan 1 dgn setpoint 20.000 Rpm dgn gangguan	61
Gambar 4.23. Percobaan 2 dgn setpoint 20.000 Rpm dgn gangguan	61
Gambar 4.24. Hasil dari tiga buah metoda (1).....	62
Gambar 4.25. Hasil dari tiga buah metoda (2).....	62
Gambar 4.26. Hasil dari tiga buah metoda (3).....	63
Gambar 4.27. Grafik efisiensi metoda GA	67

DAFTAR TABEL

Tabel.2.1. Perbedaan jaringan syaraf dan komputer.....	10
Tabel 3.1. konversi Ut ke Data Ladder.....	42
Tabel 4.1 Pengujian Ladder R-2R tanpa motor.....	51
Tabel 4.2. Pengukuran sumbu X dengan ketelitian 10 mm.....	52
Tabel 4.3. Pengukuran sumbu X dengan ketelitian 20 mm.....	52
Tabel 4.4. Pengukuran sumbu X dengan ketelitian 30 mm.....	53
Tabel 4.5. Pengukuran sumbu X dengan ketelitian 50 mm.....	53
Tabel 4.6. Pengukuran sumbu X dengan ketelitian 100 mm.....	53
Tabel 4.7. Pengukuran sumbu Y dengan ketelitian 10 mm.....	54
Tabel 4.8. Pengukuran sumbu Y dengan ketelitian 20 mm.....	55
Tabel 4.9. Pengukuran dengan ketelitian 30 mm	55
Tabel 4.10. Pengukuran sumbu Y dengan ketelitian 50 mm.....	55
Tabel 4.11. Pengukuran sumbu Y dengan ketelitian 100 mm.....	55
Tabel 4.12 Data percobaan dengan Jumlah lubang 0 s/d 49	64
Tabel 4.13 Data percobaan dengan Jumlah lubang 50 s/d 132	65
Tabel 4.14 Data percobaan dengan Jumlah lubang 133 s/d 741	66

BAB I

PENDAHULUAN

I.1 Latar belakang

Banyak riset yang telah dilakukan berkaitan dengan permasalahan proses mengebor PCB. Beberapa algoritma telah dilakukan baik secara teori maupun implementasi untuk mengatasi permasalahan posisi hole yang acak dan mendapatkan waktu terpendek agar dapat melewati semua posisi hole tersebut. Tujuan akhir yang lain dari mengebor adalah menjaga agar mata bor tidak mudah patah.

Menjaga agar mata bor tidak mudah patah, menjadi topik bahasan tersendiri dari banyak riset, mulai dari menjaga kestabilan putaran motor dengan mengatur arus/tegangan pada motor listrik dapat menjaga agar putaran motor listrik tersebut menjadi stabil, proses ini dapat dilakukan secara manual. Dalam menjaga kestabilan putaran motor listrik menggunakan NNPT (Neural Network Parameter Tune) menggunakan metoda close loop system, pengaturan parameter Kp dan Ki dapat menjaga putaran motor tersebut.

I.2 Perumusan Masalah

Pada tesis yang berjudul pengaturan pergerakan robot untuk aplikasi pengeboran PCB terdapat tiga jenis mekanik, yaitu mekanik untuk pergerakan sumbu X, pergerakan sumbu Y dan pada sumbu Z. Mekanik pada sumbu X akan mendapat beban mekanik sumbu Y dan Z, mekanik sumbu Y mendapat beban mekanik sumbu Z dan mekanik sumbu Z dapat bergerak naik dan turun agar dapat melubangi PCB. Karena mekanik sumbu X dan Y mendapat beban yang berat maka pondasi pada mekanik ini diharapkan kuat dan dapat bergerak secara halus. Penggunaan dan penempatan motor stepper merupakan permasalahan tersendiri pada mekanik sumbu

X dan Y. Pada sumbu Z, permasalahan yang timbul adalah bagaimana agar motor DC agar dapat naik dan turun secara cepat agar waktu pengeboran dapat berjalan cepat. Setelah mekanik untuk tiap tiap sumbu dapat direalisasikan langkah selanjutnya adalah mengetahui posisi koordinat dari lubang pada PCB, posisi lubang pada PCB dapat diketahui dengan mengkonversi file dari protel ke koordinat. Koordinat tersebut yang akan menjadi acuan untuk pergerakan robot. Urutan posisi koordinat yang didapat dari konversi file protel masih acak, jika robot bergerak mengikuti urutan posisi koordinat yang dihasilkan dari file protel tersebut maka pergerakan robot menjadi tidak optimal. Agar pergerakan robot menjadi optimal diperlukan metoda genetik. Metoda ini telah dikenal dapat menyelesaikan permasalahan perencanaan lintasan secara optimal. Dengan menggunakan metoda genetik diharapkan lintasan yang akan dilalui robot menjadi optimal. Setelah melakukan proses perencanaan lintasan, robot dapat bergerak menuju koordinat yang telah ditentukan sesuai dengan urutan koordinat. Proses selanjutnya adalah melubangi/pengeboran pada PCB, hasil dari pengeboran adalah lubang. Kualitas lubang yang dihasilkan tergantung dari kestabilan kecepatan. Metoda yang digunakan agar dapat menjaga kestabilan kecepatan putaran mata bor adalah NNPT (Neural Network Parameter Tune). Metoda ini merupakan pengabungan antara metoda neural network dan PI. Pada metoda PI diperlukan pengaturan Kp (Konstanta proporsional) dan KI (konstanta Integral). Yang mana pengaturan Kp dan Ki tersebut akan dilakukan oleh neural network.

1.3 Tujuan penelitian

Tujuan dari penelitian ini adalah dapat memaksimalkan waktu kerja dari proses bor PCB yang menggunakan metoda genetik dan menghasilkan lubang yang

bagus dengan menjaga kestabilan kecepatan putaran mata bor dengan menggunakan metoda NNPT.

I.4 Manfaat penelitian

Manfaat secara umum penelitian ini adalah memberikan :

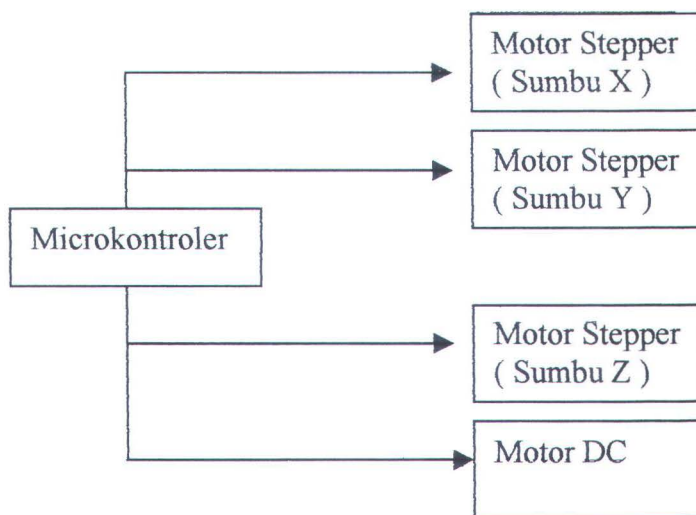
1. Implementasi dari robot kartesian , yang terdapat tiga buah koordinat X,Y dan Z.
2. Penerapan metoda genetik sebagai perencanaan lintasan agar diperoleh jarak terpendek agar robot dapat bergerak secara optimal.
3. Rincian design sensor posisi dan sensor kecepatan putaran motor DC dapat berguna bagi kelanjutan penelitian mengenai mesin bor.
4. Penerapan metoda NNPT untuk menjaga kestabilan putaran motor DC, agar diperoleh lubang yang berkualitas.

Manfaat secara khusus dari hasil penelitian ini adalah memberikan “ gambaran pemecahan permasalahan pada mesin bor PCB “

BAB II

DASAR TEORI

Pengaturan pergerakan robot untuk aplikasi pengeboran PCB terdapat beberapa macam perangkat keras yang meliputi motor DC, motor stepper dan perangkat lunak yang meliputi algoritma genetik dan *neural network parameter tune* (NNPT). Secara umum dapat digambarkan seperti pada diagram blok dibawah ini



Gambar 2.1. Diagram blok konstruksi elektronik

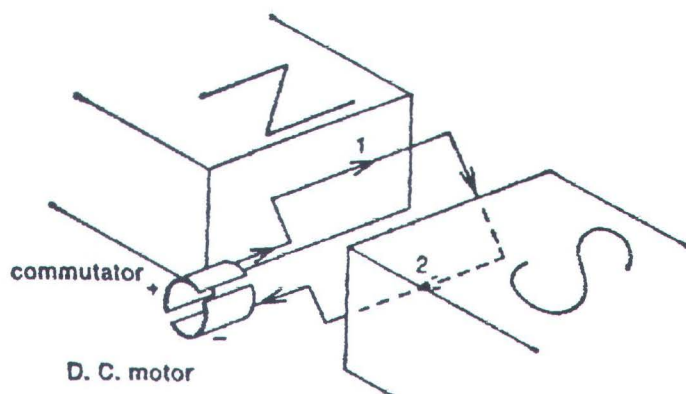
2.1. MOTOR DC.

Pada prinsipnya sebuah mesin listrik dapat berlaku sebagai motor maupun sebagai generator. Perbedaannya hanya terletak dalam konversi dayanya. Generator adalah suatu mesin listrik yang mengubah daya masuk mekanik menjadi daya keluaran listrik, sedangkan sebaliknya motor adalah mesin listrik yang mengubah daya masuk listrik menjadi daya keluaran mekanik. Konstruksi motor DC sangat

mirip dengan generator DC, dimana pada kenyataannya mesin yang bekerja dengan baik pada generator akan bekerja dengan baik pula jika sebagai motor.

2.1.1 Prinsip Kerja Motor DC.

Motor DC atau yang sering disebut juga dengan motor arus searah menjadi tenaga gerak atau tenaga mekanik, dimana tenaga gerak tersebut berupa putaran dari rotor. Prinsip kerjanya hampir sama dengan generator DC, perbedaannya hanya terletak dalam mengkonversi dayanya. Dimana prinsip dasarnya adalah sebagai berikut “apabila suatu kawat berarus diletakkan diantara kutub-kutub magnet (U-S), maka pada kawat tersebut akan bekerja suatu gaya yang akan menggerakkan kawat tersebut”. Dibawah ini merupakan gambar model prinsip kerja dari motor DC.



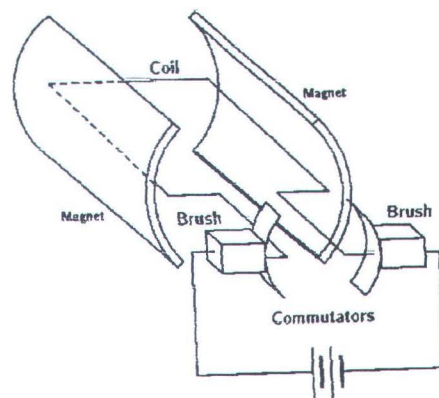
Gambar 2.2. Prinsip kerja Motor DC¹

Bila sebuah belitan terletak diantara medan magnet yang homogen, arah gerakannya seperti ditunjukkan pada gambar diatas. Hal ini dikarenakan kedua sisi belitan mempunyai arus yang arahnya berlawanan.

¹ [Http://labweb.mech.nwu.edu/mechatronics/design_ref/motor_files/motor1.gif](http://labweb.mech.nwu.edu/mechatronics/design_ref/motor_files/motor1.gif)

2.1.2. Konstruksi Motor DC.

Bagian-bagian penting dari motor DC ditunjukkan seperti pada gambar. Statornya mempunyai kutub menonjol dan diteral oleh satu atau lebih kumparan medan. Pembagian fluks celah udara yang dihasilkan oleh lilitan medan secara simetris berada disekitar garis kutub medan, sumbu ini dinamakan sumbu langsung.



Gambar 2.3. Konstruksi Motor DC²

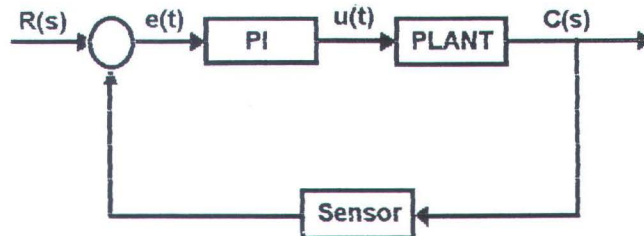
Kumparan penguat dihubungkan secara seri. Jangkar merupakan besi berlaminasi yang bergerak untuk mengurangi arus Eddy. Letak kumparan jangkar pada slot besi disebelah luar permukaan jangkar. Pada jangkar ini terdapat komunitator yang berbentuk silinder dan diisolasi. Sisi kumparan pada beberapa bagian yang berbeda, sesuai dengan jenis belitannya.

2.2. KONTROLER PROPORSIONAL INTEGRAL.

Kontroler jenis ini merupakan kombinasi dari dua jenis sistem kontrol yaitu Proportional dan Integral. Jika kedua kontroler tersebut berdiri sendiri, maka hasil yang dicapai kurang begitu bagus, dikarenakan masing-masing memiliki kelebihan dan kelemahan masing-masing, oleh karena itu, maka dikombinasikannya kedua

sistem kontrol tersebut menjadi satu sistem kontrol tunggal, yang diharapkan dapat mengeleminasikan kelemahan masing-masing dan mampu memberikan kontribusi dan kelebihanannya masing-masing.

Contoh dari PI kontroler dapat dilihat dari gambar berikut dibawah ini :



Gambar 2.4. Diagram blok PI

keterangan :

$R(s)$ = setpoint (input)

$e(t)$ = error

$u(t)$ = sinyal kontrol

$C(s)$ = output

Sebagaimana yang ditunjukkan oleh gambar tersebut, error $e(t)$ atau kesalahan merupakan selisih antara output dan setpoint, yang diproses oleh kontroler PI untuk selanjutnya menghasilkan sinyal kontrol $u(t)$ yang dikirim ke sistem atau plant untuk dikontrol.

2.2.1. Kontroler PI.

Kontroler Proporsional-Integral (PI) memiliki persamaan fungsi alih sebagai penjumlahan dari gain proporsional dan gain integral atau $G_{pi} = G_p + G_i$. Fungsi alihnya adalah sebagai berikut :

² [Http://labweb.mech.nwu.edu/mechatronics/design_ref/motor_files/motor3.gif](http://labweb.mech.nwu.edu/mechatronics/design_ref/motor_files/motor3.gif)

$$G_c(s) = K_p \left(1 + \frac{1}{T_i s} \right)$$

$$u(t) = K_p \cdot e(t) + K_i \int e(t) dt \dots \dots \dots (2.1)$$

Kontroler PI dapat membuat pengaruh *off-set* dan gangguan pada keadaan tunak masing-masing sama dengan 0 (nol). Karena itu keadaan kerja *steady state* kontroler PI lebih memuaskan. Tetapi di pihak lain kontroler PI memiliki kelemahan yang cenderung tidak stabil karena orde fungsi alih menjadi lebih besar dari pada sistem kontroler Proporsional.

Secara umum fungsi dari masing-masing kontroler dalam PI-kontroler adalah :

- Proporsional

Berfungsi untuk mempercepat terjadinya respon terhadap sinyal error, bekerja efektif pada daerah sebelum sistem mencapai setpoint (kondisi awal)

- Integral

Berfungsi memelihara sinyal kontrol menjadi konstan, bekerja efektif pada daerah dimana sistem mencapai setpoint.

2.3. JARINGAN SYARAF (NEURAL NETWORK).

2.3.1. Pendahuluan.

Sejak studi yang dilakukan oleh McCulloch dan Pitts tahun 1943 untuk menggali potensi dan kemampuan komponen yang saling berhubungan berbasis model neuron biologis, kemudian banyak peneliti yang mengembangkan apa yang dikenal dengan jaringan syaraf tiruan (JST) atau *Artificial Neural Network* (ANN).

Tidak seperti komputer konvensional yang diprogram untuk melakukan tugas-tugas tertentu yang spesifik, Jaringan Syaraf Tiruan (JST) adalah belajar atau berlatih untuk menangani tugas-tugas yang dibebankan kepadanya. Kita tidak

memilih sekumpulan instruksi dari JST untuk dieksekusi agar dapat melaksanakan tugasnya melainkan kita memilih arsitektur jaringan yang sesuai, memilih pembobot atau pemberat awal yang sesuai dan menentukan algoritma pembelajaran yang memungkinkan JST mampu mempelajari pola dan asosiasi masalah yang dihadapi.

Jaringan syaraf merupakan jaringan neuron, synapses, axon dan dendrites yang sangat kompleks. Paling tidak ada 2×10^{10} neurons, yang masing-masing memiliki 10.000 synapses yang tersebar pada masing-masing pohon dendrites yang memiliki synapses di axon dari satu neuron sebesar 10.000 juga. Pemodelan struktur ini secara rinci tentu saja akan merupakan sesuatu yang sangat tidak mungkin (setidaknya sampai saat ini), akan tetapi untuk keperluan rekayasa tidak diperlukan sebanyak itu. Sehingga dikembangkanlah arsitektur-arsitektur yang lebih sederhana, dengan tetap memakai prinsip jaringan syaraf tetapi mampu memecahkan persoalan yang cukup rumit diantaranya termasuk Error Back-propagation, Hopfield, Perceptron, CMAC, Kohonen, Adaptive Resonance Theory (ART) dan Radial Basis Function (RBF). Diantara jaringan tersebut, error back-propagation atau secara singkat dikenal sebagai backpro yang paling banyak digunakan, sehingga dalam pembahasan tugas akhir ini hanya jaringan dengan arsitektur errorback-propagation (backpro) ini saja yang akan kami bahas.



2.3.2. Model Jaringan Syaraf.

Otak manusia mengandung sekitar 10 miliar jaringan syaraf, atau dikenal sebagai neuron. Secara rata-rata masing-masing neuron dihubungkan dengan neuron lain melalui 10 ribu synapses (dapat berbeda, tergantung anatomi local neuron-nya). Jaringan neuron otak membentuk sistem pemrosesan informasi yang paralel secara

massif. Hal ini berbeda dengan komputer konvensional, karena prosesor tunggal mengeksekusi instruksi tunggal secara seri.

Dengan mempertimbangkan waktu yang dibutuhkan untuk masing-masing operasi elementer, neuron biasanya beroperasi dengan laju maksimum 100 Hz, sementara CPU komputer dapat melakukan operasi perhitungan ratusan juta eksekusi per detik. Meski pun otak dibuat dengan perangkat keras yang sangat lambat namun memiliki kemampuan yang luar biasa. Perbandingan ini dapat dilihat pada tabel berikut :

Tabel.2.1. Perbedaan jaringan syaraf dan komputer.

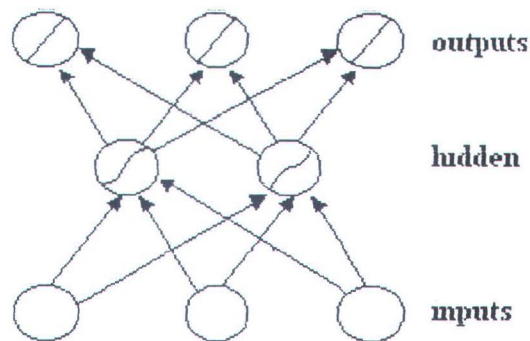
	Elemen Pemroses	Ukuran Elemen	Energi Yang Dibutuhkan	Kecepatan Proses	Cara Komputasi	Toleransi Kesalahan	Belajar	Cerdas Kesadaran
	10^{14} Sinapsis	10^{-6} m	30 W	100 Hz	Paralel, terdistribusi	Ya	Ya	Selalu
	10^8 Transistor	10^{-6} m	30 W	10^9 Hz	Serial, terdistribusi	Tidak	Sedikit	Belum

Akan tetapi untuk keperluan aplikasi praktis tentu saja tidak (atau lebih tepat belum) memerlukan jaringan syaraf serumit dan secanggih otak manusia.

Pada jaringan linier (*linier network*) dengan metode gradien descent ada target untuk masukan dan keluarannya, sedangkan untuk jaringan lapisan jamak (*multi-layer network*) akan mengalami kesulitan, karena tidak ada nilai target untuk satuan tersembunyi (*hidden units*).

Dengan menggunakan algoritma error backpropagation atau yang dikenal dengan sebagai backpro, memungkinkan cara untuk pembelajaran dengan satuan tersembunyi dalam jumlah berapa pun. Pada dasarnya jaringan tidak harus diorganisir dalam lapisan-lapisan, bentuk hubungan yang menghubungkan antar node dari keluaran ke masukan diperbolehkan. Jaringan ini tidak memperbolehkan adanya

sirkus balik, karena itu jaringan ini dikenal sebagai jaringan umpan-maju (*feedforward network*).



Gambar 2.5. Jaringan Lapis Jamak (Multi-layer Network).³

Jaringan syaraf dapat didefinisikan sebagai interkoneksi syaraf-syaraf (*neuron*), yaitu syaraf (*neuron*) sebagaimana dalam persamaan :

$$o = f(w^t x) \text{ atau } o = f\left(\sum_{i=1}^n w_i x_i\right)$$

$$w \overset{\Delta}{=} [w_1 \ w_2 \ \dots \ w_n]^t$$

$$x \overset{\Delta}{=} [x_1 \ x_2 \ \dots \ x_n]^t \dots\dots\dots(2.2)$$

Keterangan : w = vektor berat

x = input vektor

atau :

$$f(net) \overset{\Delta}{=} \frac{1}{1 + \exp(-\lambda net)} \text{ dan } f(net) \overset{\Delta}{=} \begin{cases} 1, & net > 0 \\ 0, & net < 0 \end{cases}$$

$$f(net) \overset{\Delta}{=} \frac{2}{1 + \exp(-\lambda net)} - 1 \text{ dan } f(net) \overset{\Delta}{=} \text{sgn}(net) = \begin{cases} +1, & net > 0 \\ -1, & net < 0 \end{cases} \dots\dots\dots(2.3)$$

³ Introduction to artificial neural systems ,hal 38

yang output-outputnya dihubungkan melalui bobot W keseluruhan syaraf termasuk syaraf utama yang melibatkan penurunan waktu delay.

2.3.2.1. Jaringan Umpanmaju (Feedforward Network).

Arsitektur dari umpanmaju direferensikan dengan m neuron dan n input sebagaimana ditunjukkan pada gambar (2.9), yang menunjukkan bahwa input dan output vektornya adalah :

$$\begin{aligned} o &= [o_1 \ o_2 \ \dots \ o_m]^t \\ x &= [x_1 \ x_2 \ \dots \ x_n]^t \end{aligned} \quad \dots \dots \dots (2.4)$$



Bobot atau pemberat W_{ij} dihubungkan dengan neuron i dengan input j . Kita dapat menuliskan nilai aktivasi untuk neuron i dengan persamaan :

$$net_i = \sum_{j=1}^n w_{ij} x_j, \quad \text{untuk } i = 1, 2, \dots, m \quad \dots \dots \dots (2.5)$$

Transformasi persamaan linier termasuk fungsi aktivasi $f(net_i)$, dimana $i = 1, 2, \dots, m$, guna melengkapi proses pada persamaan x . Transformasi yang ditunjukkan oleh setiap neuron m didalam jaringan ditunjukkan dengan pemetaan nonlinier berikut :

$$o_i = f(w_i^t x_i) \quad \text{untuk } i = 1, 2, \dots, m \quad \dots \dots \dots (2.6)$$

dimana vektor berat W_i berisi berat yang menunjukkan output i , sesuai dengan persamaan :

$$W_i = [w_{i1} \ w_{i2} \ \dots \ w_{in}]^t \quad \dots \dots \dots (2.7)$$

Matrik operator nonlinier Γ , input pemetaan x ke output o di implementasikan dalam jaringan dengan persamaan :

$$o = \Gamma [Wx] \quad \dots \dots \dots (2.8)$$

dimana w adalah matrik berat, yang juga dikenal dengan nama matrik koneksi :

$$W = \begin{bmatrix} w_{11} & w_{12} & \dots & w_{1n} \\ w_{21} & w_{22} & \dots & w_{2n} \\ \vdots & \vdots & \dots & \vdots \\ w_{m1} & w_{m2} & \dots & w_{mn} \end{bmatrix} \dots\dots\dots (2.9)$$

dan

$$\Gamma[\cdot] = \begin{bmatrix} f(\cdot) & 0 & \dots & 0 \\ 0 & f(\cdot) & \dots & 0 \\ \vdots & \vdots & \dots & \vdots \\ 0 & 0 & \dots & f(\cdot) \end{bmatrix} \dots\dots\dots (2.10)$$

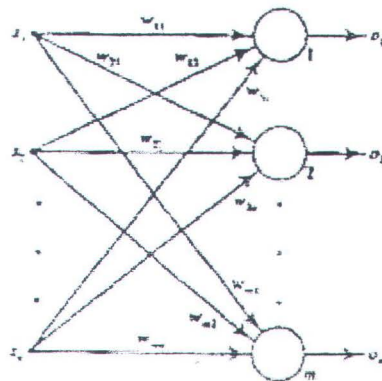
Dari persamaan diatas terdapat fungsi aktivasi nonlinier $f(\cdot)$ diatas diagonal dari matrik operator Γ pada nilai aktivasi *net* dari setiap neuron. Setiap nilai aktivasi berbentuk perkalian skalar dari setiap input dengan vektor berat.

Input x dan output o sering disebut dengan persamaan input-output. Pemetaan dari persamaan input ke persamaan output sebagaimana ditunjukkan persamaan (x) diatas merupakan umpanbalik. Sehingga didalamnya tidak terdapat time delay antara input x dan output o . Maka persamaanya dapat disederhanakan menjadi :

$$o(t) = \Gamma [Wx(t)] \dots\dots\dots (2.11)$$

Pada gambar (2.9) menunjukkan jaringan umpan maju. Tipe dari jaringan ini dapat dihubungkan dalam keadaan cascade untuk membuat jaringan multilayer (*multilayer network*). Sebagaimana dalam setiap jaringan, output dari layer akan menjadi input bagi layer berikutnya. Jaringan umpanmaju tidak memiliki koneksi umpan balik yang sederhana ketika $x(t)$ dipetakan kedalam $o(t)$, nilai outputnya

sering dianalogikan sebagai “sumber input”, yang menyediakan nilai output tertentu dan sinyal error yang dihasilkan guna mengadaptasikan berat jaringan.



Gambar 2.6. Jaringan Umpan Maju Single-Layer⁴

2.3.2.2. Jaringan Umpanbalik (Feedback Network).

Jaringan umpan-balik dapat diperoleh dari jaringan umpan-maju. Dengan menghubungkan output *neuron* ke inputnya sehingga hasilnya dapat dilihat pada gambar (2.7). Inti dari umpan-balik tertutup ini adalah untuk mengaktifkan kontrol output O_i melalui output O_j , dimana $j = 1, 2, \dots, m$.

Sebagaimana dengan kontrol yang memiliki peran penting dalam menghasilkan output, misalnya $o(t)$ adalah pengontrol output dari persamaan $o(t + \Delta)$. Waktu Δ berkisar antara $t - (t + \Delta)$ yang dikenal dengan delay dalam loop umpan-balik sebagaimana seperti yang ditunjukkan dalam gambar (2.10), dengan menggunakan notasi pada jaringan umpan-maju, maka pemetaan dari $o(t)$ ke $o(t + \Delta)$ dapat ditulis dengan persamaan :

$$o(t + \Delta) = \Gamma [Wo(t)] \dots\dots\dots(2.12)$$

⁴ Introduction to artificial neural systems ,hal 40

Input $x(t)$ diperlukan untuk menginisialisasi jaringan sehingga $o(0) = x(0)$.

Inputnya kemudian dipindahkan sehingga sistem menjadi $t > 0$. Kita harus memperhatikan konfigurasi dari feedback ini, seperti $x(t) = x(0)$ dan tidak ada input yang digunakan untuk jaringan tersebut pada kondisi $t > 0$.

Jika kita menganggap waktu sebagai variabel diskrit untuk menganalisa penampakan jaringan pada rentang waktu diskrit $\Delta, 2\Delta, \dots$, sistem ini disebut waktu diskrit. Untuk aturan notasi waktu dari jaringan ini dalam bentuk satuan, dan bentuk waktu ini berupa bilangan bulat positif, $(\Delta) = \text{satuan delay}$. Untuk waktu diskrit dari jaringan syaraf tiruan pada persamaan $o(t + \Delta) = \Gamma [W_o(t)]$ dikonversikan menjadi persamaan :

$$o^{k+1} = \Gamma [W_o^k] \dots\dots\dots (2.13)$$

untuk $k = 1, 2, \dots$

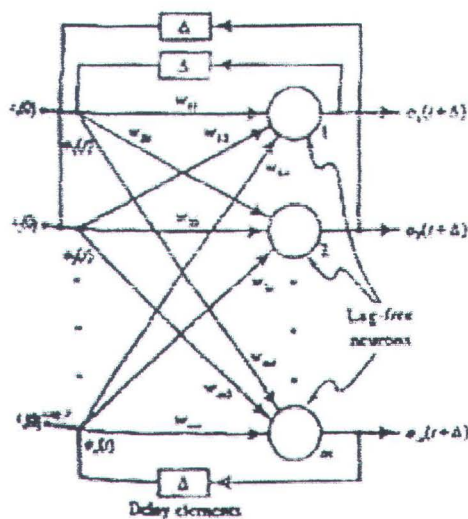
Jaringan pada gambar (2.7) disebut jaringan *recurrent* karena responnya terjadi pada $(k + 1)$ dimana jaringan dimulai dari $k = 0$, sehingga didapatkan persamaan :

$$\begin{aligned} o^1 &= \Gamma [Wx^0] \\ o^2 &= \Gamma [W\Gamma [Wx^0]] \\ &\dots\dots\dots \\ o^{k+1} &= \Gamma [W\Gamma [\dots\Gamma [Wx^0] \dots]] \dots\dots\dots (2.14) \end{aligned}$$

Tipikal jaringan *recurrent* dioperasikan dengan representasi dari sebuah data diskrit, dengan menggunakan neuron-neuron pada limit fungsi aktivasi. Sistem dengan input waktu diskrit dan representasi dari data diskrit diatas dinamakan

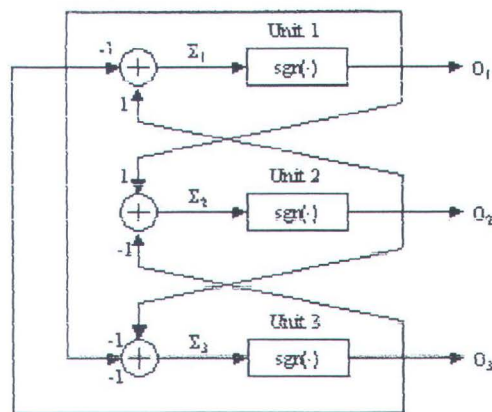
automaton. Sehingga jaringan syaraf *recurrent* pada jenis ini dinamakan automaton-automaton.

Persamaan $o^{k+1} = \Gamma [W\Gamma [\dots \Gamma [Wx^0] \dots]]$ dinamakan keadaan o^k dari jaringan dimana $k = 1, 2, 3, \dots$, dan menghasilkan keadaan transisi. Selanjutnya jaringan dimulai dari keadaan transisi pada kondisi awal (input = 0) atau x^0 , hingga pada keadaan transisi o^k , dimana $k = 1, 2, \dots$, hingga membentuk keadaan equalibrium. Keadaan equalibrium ini dinamakan attractor. Sebuah attractor dapat berupa suatu keadaan dengan nilai tertentu. Contoh jaringan sederhana dari jaringan syaraf seperti terlihat pada gambar (2.8) menggambarkan attractor dari suatu keadaan dengan nilai $o = [1 \ 1 \ -1]^T$.



Gambar 2.7. Jaringan Umpan Balik Waktu Diskrit⁵

⁵ Introduction to artificial neural systems ,hal 42



Gambar 2.8. Jaringan Syaraf Sederhana⁶

2.3.3. Algoritma Backpropagation.

Pembelajaran jaringan umpan-maju lapisan banyak (*multi-layer network*) dengan menggunakan metode gradient descent untuk mendekati suatu fungsi yang tidak diketahui, berdasarkan pada beberapa data pembelajaran yang terdiri dari pasangan vektor (x, t) . Vektor x mewakili pola masukan jaringan, dan vektor t merupakan target atau keluaran yang diinginkan. Gradien keseluruhan untuk masing-masing himpunan training adalah merupakan jumlah gradien dari masing-masing pola. Masing-masing satuan akan diberi nomor dan pembobot (*wieght*) dari satuan j ke unit i diberi notasi w_{ij} .

Untuk jaringan umpan maju yang berlapis-lapis (*multi-layer*) dan dapat dihubungkan secara penuh, yaitu masing-masing node pada jaringan tertentu terhubung dengan setiap node pada jaringan berikutnya, maka algoritmanya akan lebih mudah ditulis dalam bentuk atau notasi matriks. Pada notasi ini, bias pemberat (bobot bias), masukan, aktivasi, dan sinyal error untuk semua satuan di suatu lapisan dikombinasikan ke dalam vektor, sementara pemberat non-bias dari satu lapis ke

⁶ Introduction to artificial neural systems, hal 46

lapis berikutnya membentuk matriks W. Algoritma backpronya secara umum dapat dituliskan sebagai berikut :

1. Inisialisasi lapisan masukan :

$$\vec{y}_0 = \vec{x} \dots\dots\dots(2.15)$$

2. Mempropagasikan aktifitas maju untuk $l = 1, 2, \dots, L$,

$$\vec{y}_l = f_l(W_l \vec{y}_{l-1} + \vec{b}_l) \dots\dots\dots(2.16)$$

dimana b_l adalah pemberat bias.

3. Menghitung error di lapisan keluaran :

$$\vec{\delta}_L = \vec{t} - \vec{y}_L \dots\dots\dots(2.17)$$

4. Propagasi balik errornya, dari $l = L-1, L-2, \dots, 1$.

$$\vec{\delta}_l = (W_{l+1}^T \vec{\delta}_{l+1}) \cdot f_l'(\vec{net}_l) \dots\dots\dots(2.18)$$

dimana T adalah matriks operator transposisi.

5. Update pemberat dan bias :

$$\Delta W_l = \vec{\delta}_l \vec{y}_{l-1}^T \quad \Delta \vec{b}_l = \vec{\delta}_l \dots\dots\dots(2.19)$$

2.3.3.1. Definisi.

- Sinyal error untuk satuan j : $\delta_j = -\partial E / \partial net_j$.
- Gradien (negatif) untuk pembobot w_{ij} : $\Delta w_{ij} = -\partial E / \partial w_{ij}$.
- Sekumpulan nodes anterior ke satuan i : $A_i = \{j : \exists w_{ij}\}$.
- Sekumpulan nodes posterior ke satuan j : $P_j = \{i : \exists w_{ij}\}$.

2.3.3.2.Gradient.

Seperti halnya jaringan linier (*linier Network*), kita dapat melakukan ekspansi gradien menjadi dua faktor dengan menggunakan dalil rantai :

$$\Delta w_{ij} = - \frac{\partial E}{\partial net_i} \frac{\partial net_i}{\partial w_{ij}} \dots\dots\dots (2.20)$$

Faktor pertama adalah error dari satuan i dan diberi notasi δ_i , sedangkan yang kedua adalah :

$$\frac{\partial net_i}{\partial w_{ij}} = \frac{\partial}{\partial w_{ij}} \sum_{k \in A_i} w_{ik} y_k = y_i \dots\dots\dots (2.21)$$

Sehingga bila digabungkan akan diperoleh persamaan :

$$w_{ij} = \delta_i y_i \dots\dots\dots (2.22)$$

Untuk menghitung gradien ini, perlu diketahui aktivitas dan error untuk seluruh node di jaringan.

2.3.3.3. Aktivitas Maju (Forward Activation).

Aktivitas satuan masukan ditentukan oleh masukan eksternal jaringan x. Untuk satuan yang lain, aktivitasnya dipropagasi maju :

$$y_i = f_i \left(\sum_{j \in A_i} w_{ij} y_j \right) \dots\dots\dots (2.23)$$

Sebelum aktivitas unit 1 dihitung, seluruh aktivitas node sebelumnya (anterior node, yang membentuk himpunan A_i) harus diketahui. Karena jaringan umpan-maju tidak mengandung siklus atau umpan balik, maka hal ini dapat dihitung dengan mudah.

2.3.3.4. Menghitung Error Keluaran (Calculating output error).

Diasumsikan bahwa rugi-rugi jumlah-kuadrat (sum-squared loss) yang digunakan, yaitu :

$$E = \frac{1}{2} \sum_o (t_o - y_o)^2 \dots\dots\dots (2.24)$$

Error untuk satuan keluaran dapat dihitung dengan menggunakan rumus :

$$\delta_o = t_o - y_o \dots\dots\dots(2.25)$$

2.3.3.5. Error Backpropagation.

Untuk satuan tersembunyi (hidden unit), maka dilakukan propagation balik dari error node keluaran (algoritma ini dikenal dengan error backpropagation). Dengan menggunakan dalil rantai, maka error satuan tersembunyi dapat diekspansikan dalam bentuk node posterior-nya :

$$\delta_i = -\sum_{i \in P_i} \frac{\partial E}{\partial net_i} \frac{\partial net_i}{\partial y_i} \frac{\partial y_i}{\partial net_i} \dots\dots\dots(2.26)$$

Dari tiga faktor di dalam tanda jumlah, bentuk pertama adalah error dari node ke-i. Bentuk keduanya adalah :

$$\frac{\partial net_i}{\partial y_i} = \frac{\partial}{\partial y_i} \sum_{k \in A_i} w_{ik} y_k = w_{ij} \dots\dots\dots(2.27)$$

sementara untuk bentuk ketiga adalah turunan fungsi aktivasi node j

$$\frac{\partial y_i}{\partial net_i} = \frac{\partial f_i(net_j)}{\partial net_j} = f'_j(net_j) \dots\dots\dots(2.28)$$

Untuk satuan tersembunyi h yang menggunakan fungsi aktivasi tanh, maka dapat diperoleh identitas $\tanh(u)' = 1 - \tanh(u)^2$, dan

$$f'_h(net_h) = 1 - y_h^2 \dots\dots\dots(2.29)$$

Memasukan semua yang diperoleh, maka akan didapatkan :

$$\delta_j = f'_j(net_j) \sum_{i \in P_i} \delta_i w_{ij} \dots\dots\dots(2.30)$$

Untuk menghitung error di satuan j, perlu diketahui terlebih dahulu seluruh error dari posterior node (yang membentuk himpunan P_i). Kemudian, bila tidak ada

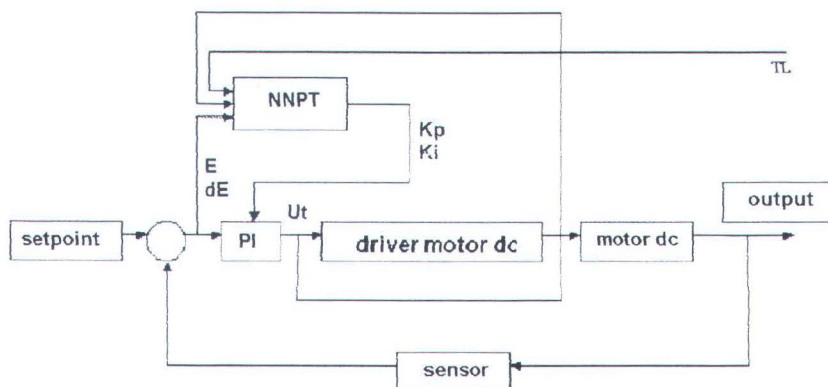
umpan-balik di jaringan maka urutan node dari keluaran kembali ke masukan yang akan memenuhi kondisi ini.

2.4. Pembelajaran Neural-Network-based Parameter Tuner (NNPT).

Sebagian besar kontroler motor DC untuk aplikasi industri menggunakan model PI fixed-gain. Meskipun tanpa model sistem yang dinamik dan mendapat gangguan, tetapi mampu menghasilkan nilai yang determinan.

Pembelajaran dan pemetaan nonlinier pokok, model Jaringan Syaraf Tiruan (JST) telah berhasil digunakan untuk beberapa sistem dan masalah kontrol. Untuk PI kontrol konvensional, disarankan menggunakan Neural-Network-Based Self-Tuning PI Control System yang dapat mengontrol gainnya sendiri (self-tune) pada beberapa kondisi operasi.

Pengontrolan atau pengaturan motor dalam hal ini adalah Motor DC, dapat dilakukan dengan menggunakan metode PI kontroler dan jaringan syaraf tiruan bertujuan untuk mengatur agar kecepatan motor tetap stabil tanpa harus mengetahui rugi-rugi yang dialami motor DC. Blok diagramnya seperti pada Gambar (2.12) berikut :



Gambar 2.9. Diagram Blok Pengaturan Motor DC dengan PI Kontroler dan JST

Keterangan :

NNPT merupakan Neural-network-based parameter tuner

PI adalah PI controller.

Kp dan Ki adalah konstanta PI.

Setpoint : Berupa masukan (kecepatan) dari user.

Driver motor DC : konversi digital to analog.

Motor DC : beban

Sensor : Berupa pembacaan data kecepatan motor.

Kemungkinan NNPT menghasilkan gain yang sesuai menurut kondisi operasi, maka susunan input Neural Network atau jaringan syaraf tiruan harus berisi variabel-variabel yang dapat menunjukkan kondisi operasi motor. Pada sistem ini digunakan 3 (tiga) variabel E_θ , De , Ut dan T_L , output dari NNPT adalah Kp dan Ki .

Sebelum neural-network-based parameter tuner (NNPT) dapat digunakan untuk proses pembelajaran, NNPT akan menjalankan intelegent parameter tune terlebih dahulu. Proses pembelajaran ini akan mulai efektif begitu output jaringan sebanding dengan output yang diinginkan. Didalam sistem pembelajaran ini terdiri dari 6 (Enam) unsur yaitu : E , De , Ut , T_L , Kp dan Ki .

Keterangan :

E , De , Ut , T_L adalah variabel input

Kp dan Ki adalah variabel output.

Struktur dari NNPT adalah jaringan syaraf umpanmaju *three-layer* dengan input, hidden, dan output. Alasan menggunakan jaringan syaraf umpan maju yaitu simpel didalam mengimplementasi dan hampir semua menggunakan model jaringan

syaraf (Neural-Network model). Algoritma pembelajaran parameter self-tuning (RSTL) secara otomatis dapat mengatur parameter pembelajaran hingga NNPT menghasilkan output.

2.5. Metoda Genetik Algoritma

Algoritma Genetika adalah algoritma yang memanfaatkan proses seleksi alamiah yang dikenal dengan proses evolusi. Dalam proses evolusi, individu secara terus-menerus mengalami perubahan gen untuk menyesuaikan dengan lingkungan hidupnya. “Hanya individu-individu yang kuat yang mampu bertahan”. Proses seleksi alamiah ini melibatkan perubahan gen yang terjadi pada individu melalui proses perkembangbiakan. Dalam algoritma genetika ini, proses perkembang-biakan ini menjadi proses dasar yang menjadi perhatian utama, dengan dasar berpikir: “Bagaimana mendapatkan keturunan yang lebih baik”.

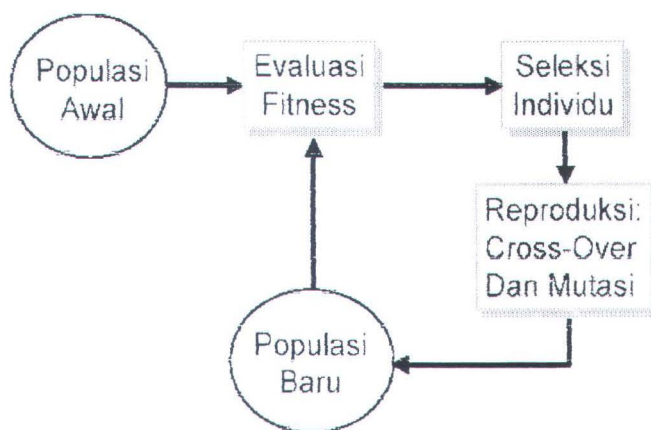
2.5.1. Definisi penting dalam Algoritma Genetika

Genotype (Gen) adalah sebuah nilai yang menyatakan satuan dasar yang membentuk suatu arti tertentu dalam sekelompok gen yang dinamakan kromosom. Dalam algoritma genetika, gen bisa berupa nilai biner, float, ataupun integer. Allele, adalah nilai suatu gen. Kromosom adalah gabungan dari gen yang membentuk nilai tertentu. Individu, menyatakan satu nilai atau keadaan yang menyatakan salah satu solusi yang mungkin dari permasalahan yang diangkat. Populasi, merupakan sekumpulan individu yang akan diproses bersama dalam satu siklus proses evolusi. Generasi, menyatakan satu-satuan siklus proses evolusi. Nilai Fitness, menyatakan seberapa baik nilai dari suatu individu atau solusi yang didapatkan.

2.5.2. Langkah langkah yang harus dilakukan dalam menggunakan algoritma genetika

Yang pertama kali yang dilakukan dalam menggunakan algoritma genetika adalah mendefinisikan individu dimana individu tersebut menyatakan salah satu penyelesaian dari permasalahan yang diangkat. Individu tersebut lalu dihitung dengan tujuan untuk mengetahui kualitas dari individu tersebut , hal ini disebut nilai *fitnes*. Selanjutnya adalah Menentukan proses pembangkitan populasi awal. Hal ini biasanya dilakukan dengan menggunakan pembangkitan acak. Langkah selanjutnya menentukan proses seleksi, proses perkawinan silang/*cros over* dan mutasi gen yang akan digunakan.

2.5.3. Algoritma genetika dalam perangkat lunak



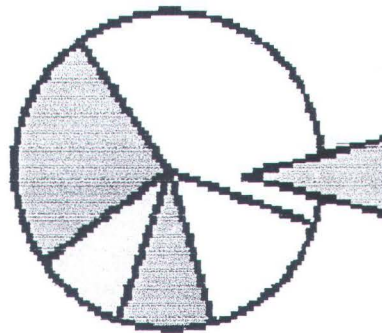
Gambar 2.10. Diagram blok Algoritma genetika

Membangkitkan populasi awal, pada permasalahan ini yang dimaksud populasi adalah hubungan antar lubang pada PCB. Hubungan antar lubang tersebut terjadi dengan cara menghubungkan secara acak / random. Dengan contoh pada PCB terdapat 8 lubang maka terdapat 9 gen (8 gen + 1 gen duplikat dari gen 1) dengan asumsi robot akan kembali ke titik awal. Langkah selanjutnya adalah evaluasi *fitnes* .

individu yang baik / berkualitas mempunyai nilai fitness yang besar sedangkan tujuan dari penggunaan algoritma genetika untuk mendapatkan lintasan terpendek.

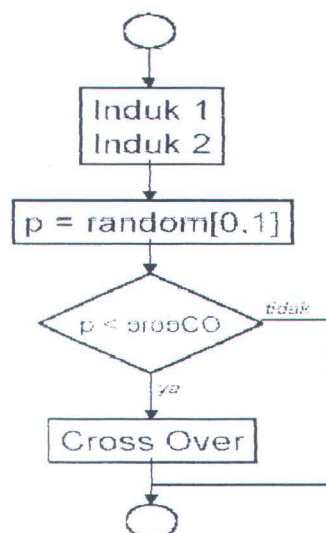
$$\text{fitness} = 1000 - \sum \sqrt{((x - x_1)^2 + (y - y_1)^2)} \dots\dots\dots(2.31)$$

setelah mendapatkan nilai fitness maka langkah selanjutnya adalah seleksi individu, seleksi ini berfungsi untuk mendapatkan calon induk yang



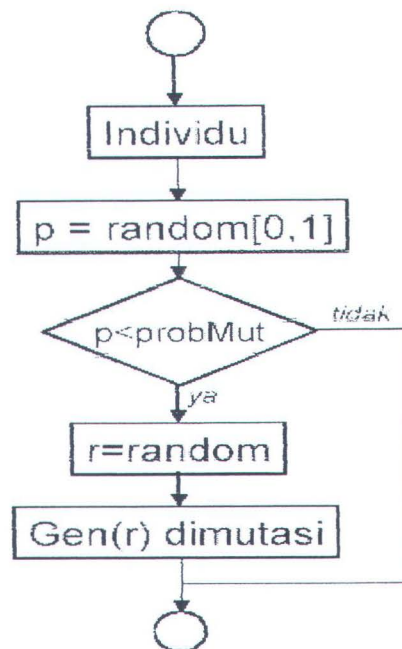
Gambar 2.11. Mesin Roulette

berkualitas dengan harapan akan menghasilkan anak yang baik. Semakin tinggi nilai fitness dari suatu individu maka semakin besar kemungkinan untuk terpilih. Seleksi ini menggunakan teknik mesin roulette. Setiap bagian dari mesin roulette ini menggambarkan besarnya nilai fitness dari setiap individu.



Gambar 2.12. Diagram blok *Crossover*

Individu yang memiliki nilai fitness besar kemungkinan terpilih semakin besar pula. Setelah mendapatkan individu yang baru maka proses selanjutnya adalah reproduksi dengan cara perkawinan sebgayaan gen dari dua buah induk / individu. Proses ini dinamakan cros over / perkawinan silang. Cros over ini akan menghasilkan individu baru yang kemungkinan lebih kuat dari induknya.



Gambar 2.13. Diagram blok mutasi Gen

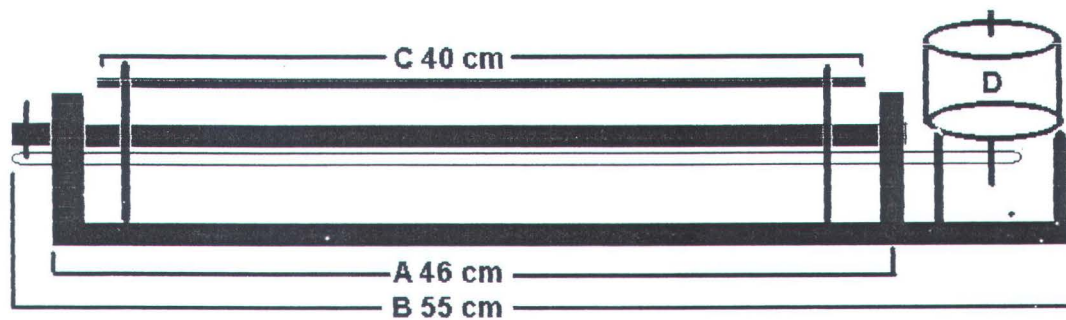
Setelah melakukan cros over langkah selanjutnya adalah mutasi gen yang akan mengeser / merubah gen gen dalam satu individu pergeseran ini ditentukan dengan probabilitas mutasi yang telah ditentukan.

BAB III

PERANCANGAN SISTEM

3.1 KONSTRUKSI MEKANIK

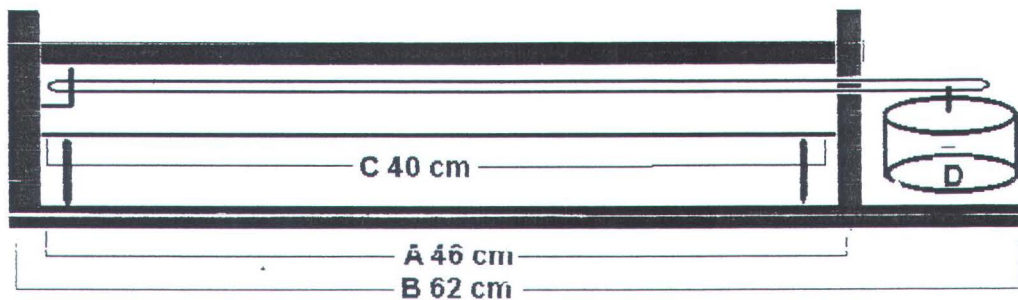
3.1.1 konstruksi mekanik sumbu x



Gambar 3.1. Konstruksi Mekanik sumbu X

Gambar 3.1. adalah gambar konstruksi mekanik sumbu X tampak depan. Titik B adalah penyangga beban, yang terbuat dari alumunium dengan ketebalan 3.5 mm dengan panjang 55 cm. Penyangga ini menompang motor stepper (D), belt (sabuk), pengaris (sensor), besi. Pada titik D adalah motor stepper 4 bit dengan tahanan dalam 20 ohm, motor stepper ini bergerak untuk 1 stepnya sebesar 0.3 mm dan jika digerakkan 0.5 step sebesar 0.15 mm, pada motor stepper diikat belt (sabuk) yang mana belt tersebut akan mengerakkan dudukan/pangkon untuk sumbu Z, panjang dari sabuk 50 cm. Titik C adalah pengaris yang berfungsi sebagai sensor pergerakan dari motor stepper, pengaris ini dengan ketelitian 10 mm untuk satu kali periode (hitam dan bening) pengukuran dapat dilakukan dengan satu periode atau setengah periode. Untuk setengah periode “menghitung banyaknya hitam dan bening” diperlukan kontruksi penyangga yang kokoh, panjangnya pengaris 40 cm agar pengaris dapat berdiri diperlukan dua buah baut untuk menyangganya.

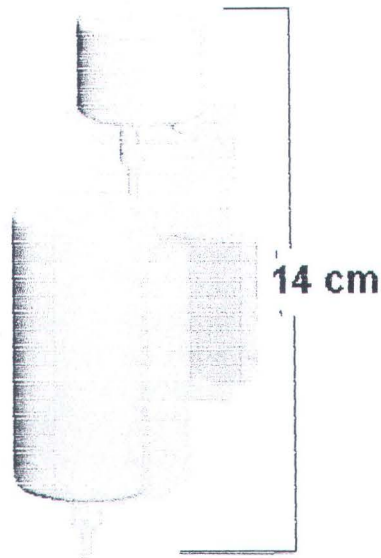
3.1.2 Konstruksi Mekanik Sumbu Y



Gambar 3.2. Konstruksi Mekanik sumbu Y

Gambar 3.2 adalah gambar konstruksi mekanik sumbu Y tampak Samping. Titik B adalah penyangga beban, yang terbuat dari alumunium dengan ketebalan 3.5 mm dengan panjang 62 cm. Penyangga ini menompang motor stepper (D), belt (sabuk), pengaris (sensor), besi, dan sumbu X. Pada titik D adalah motor stepper 4 bit dengan tahanan dalam 20 ohm, motor stepper ini bergerak untuk 1 step nya sebesar 0.3 mm dan jika digerakkan 0.5 step sebesar 0.15 mm, pada motor stepper diikat belt (sabuk) yang mana belt tersebut akan mengerakkan dudukan / pangkon untuk sumbu X, panjang dari sabuk 50 cm. Titik C adalah pengaris yang berfungsi sebagai sensor pergerakan dari motor stepper, pengaris ini dengan ketelitian 10 mm untuk satu kali perioda (hitam dan bening) pengukuran dapat dilakukan dengan satu periode atau setengah perioda. Untuk setengah periode “menghitung banyaknya hitam dan bening” diperlukan kontruksi penyangga yang kokoh, panjangnya pengaris 40 cm agar pengaris dapat berdiri diperlukan dua buah baut untuk menyangganya.

3.1.3 Konstruksi Mekanik Sumbu Z



Gambar 3.3. Konstruksi Mekanik sumbu Z

Gambar 3.3. merupakan gambar kontruksi mekanik sumbu Z tampak depan, pada sumbu z ini terdapat dua buah motor, yaitu motor DC dan motor stepper. Motor DC berfungsi untuk menggerakkan mata bor yang mana berfungsi untuk melubang PCB. Dan motor stepper berfungsi sebagai penggerak motor DC agar dapat bergerak naik atau turun. Pada sumbu Z ini tidak terdapat sensor posisi untuk mengetahui posisi naik atau turun.

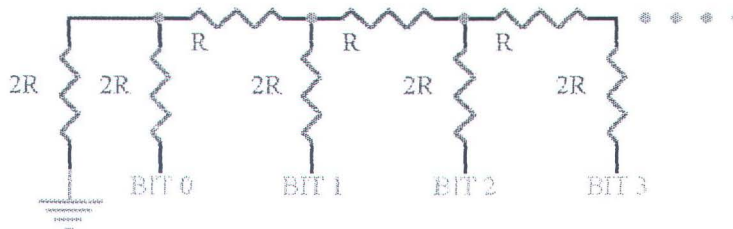
3.2 KONSTRUKSI ELEKTRONIK

3.2.1 Pengaturan Kecepatan Motor DC

Agar dapat melubangi PCB maka diperlukan sebuah motor DC yang pada ujungnya dipasang mata bor, kecepatan berputar dari mata bor dapat mempengaruhi kualitas dari lubang yang dihasilkan, agar kecepatan pada motor DC tersebut stabil maka diperlukan pengaturan tegangan pada motor tersebut. Terdapat banyak cara untuk mengatur tegangan secara digital salah satunya adalah dengan menggunakan IC

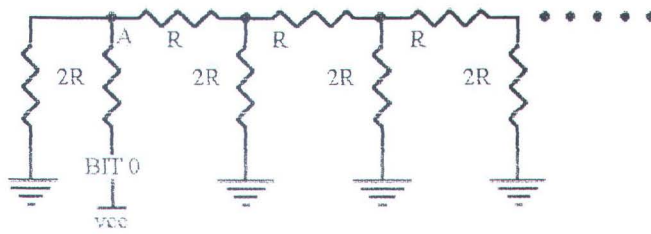
DAC (Digital Analog Converter), secara umum DAC menggunakan dua polaritas catu daya. Penggunaan dua polaritas catu daya menjadi tidak efisien dalam desain sistem ini, karena catu daya (-) hanya digunakan untuk IC DAC dan OpAmp. Agar menggunakan satu polaritas saja maka dalam desain ini menggunakan metoda dasar dari DAC. Metoda DAC terdapat berbagai macam cara. Pada desain ini menggunakan metoda R-2R ladder untuk mengatur tegangan outputnya.

Metoda R-2R ladder untuk resolusi 8 bit menggunakan 16 resistor yang dikombinasikan yang pada setiap titik 1 sampai 8. Sebagai sumber tegangan pada tiap titik didapat dari IC TTL Dengan diberikan tegangan logical (1 / 0) sebagai sumber tegangan yang mana akan menghasilkan arus (I total) yang kemudian I total tadi di rubah menjadi tegangan dengan menggunakan penguat non inverting . untuk lebih jelasnya dapat dilihat pada gambar 3.4.



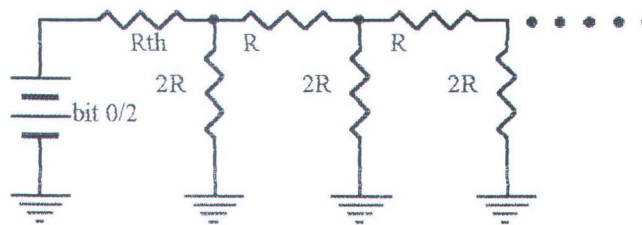
Gambar 3.4. Ladder R-2R

Gambar 3.4. adalah rangkaian Ladder R-2R untuk 4 bit, agar memudahkan perhitungan rangkaian tersebut dihitung setiap bitnya. Langkah pertama pada bit 0 yang akan diitung. Bit 0 diberi logika high dan bit yang lain diberi logika 0, seperti tampak pada gambar 3.4.

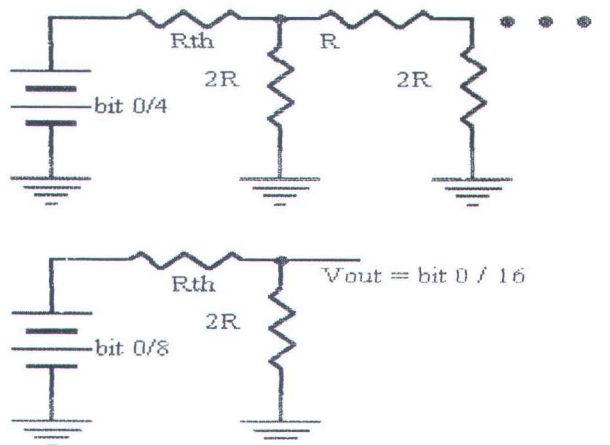


Gambar 3.5. Langkah 1.a

Tegangan pada titik A sama dengan bit 0/2, dan nilai R_{th} pada setiap bit akan selalu R .



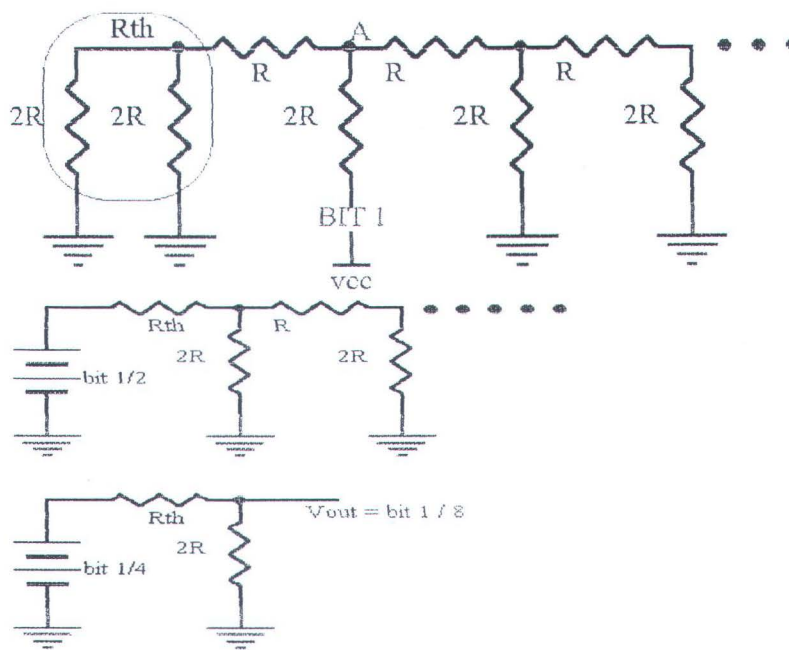
Gambar 3.6. Langkah 1.b



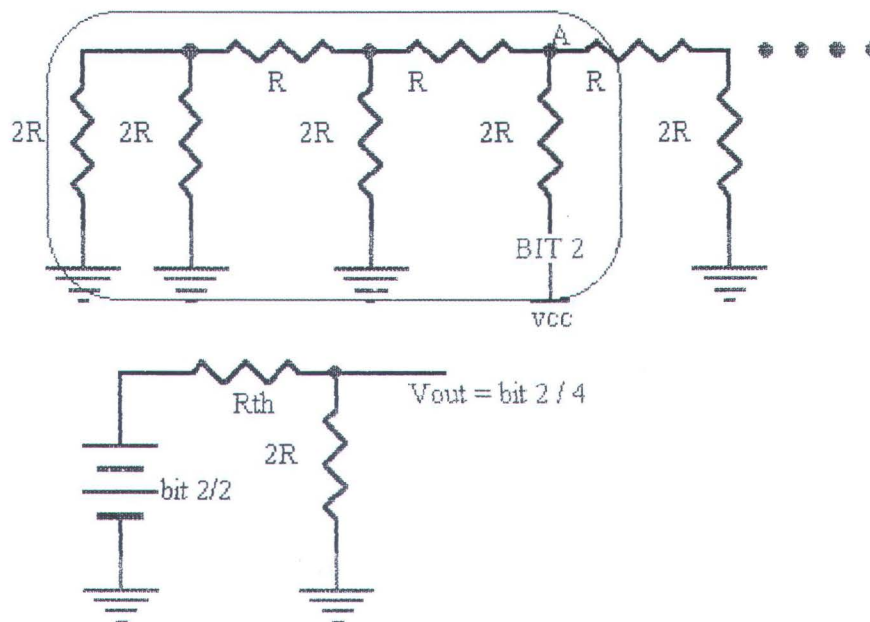
Gambar 3.7. Langkah 1.c

Pada ujung akhir bit ke 3 nilai $v_{out} = \text{bit } 0/16$

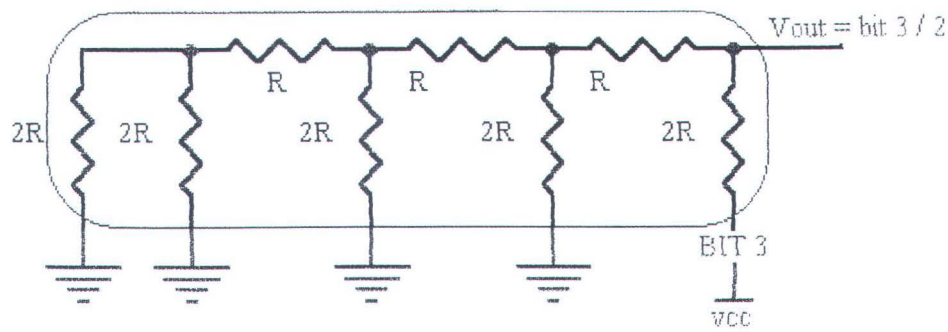
Langkah ke dua, bit 1 diberi logika high dan bit yang lain diberi logika low



Gambar 3.8. Langkah 2

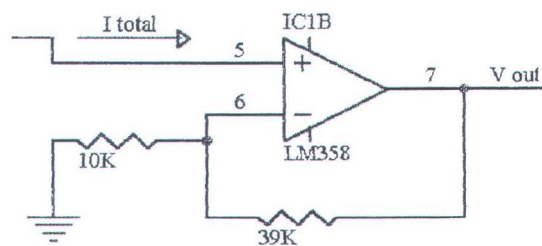


Gambar 3.9. Langkah 3



Gambar 3.10. Langkah 4

Hasil maskimal dalam keadaan high pada tiap bit adalah 5 volt sedangkan pada motor diperlukan sumber maksimal 25 volt maka output dari R ladder tersebut di kuatkan dengan menggunakan non inverting amplifier seperti pada gambar 3.11.



Gambar 3.11. Non inverting amplifier

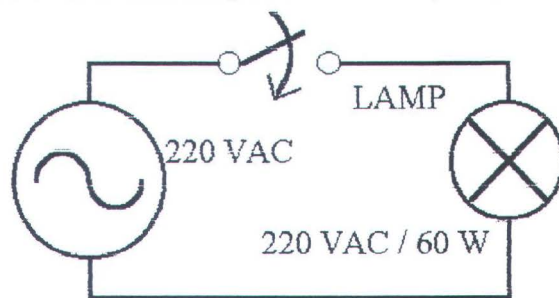
Penguat non inverting memiliki gain sebesar $(R_f / R_i) + 1 = (39k / 10k) + 1 = 4.9$ kali, maka hasil maksimal menjadi 24.5 volt. Tegangan output masih perlu dikuatkan arusnya agar mampu menggerakkan motor. Gambar rangkaian lengkap dapat dilihat pada lampiran.

3.2.2 Sensor Kecepatan

Fungsi dari mengetahui kecepatan motor DC adalah sebagai *feedback* kedalam sistem agar kecepatan motor DC tersebut dapat berputar stabil. Untuk dapat mengetahui kecepatan motor tersebut dapat menggunakan rotary encoder dan opto coupler, rotary tersebut akan berputar mengikuti gerak dari motor, berputarnya rotary encoder akan menghalangi cahaya dari optocoupler. Semakin cepat motor berputar

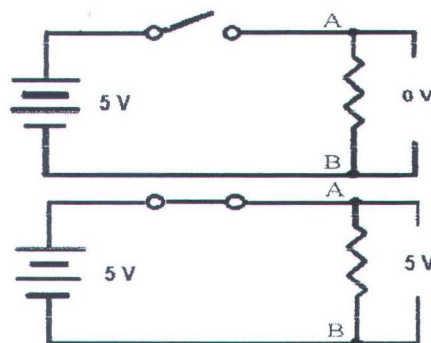
semakin sering cahaya optocoupler tersebut terkena halangan. Pengukuran kecepatan dengan menggunakan rotary encoder dan optocoupler membutuhkan tempat untuk rotary encoder pada motor, posisinya bisa di bagian atas maupun bawah. Disinilah awal dari permasalahan, karena tempat untuk memasang hanya dapat ditaruh didekat mata bor sehingga debu bekas pengeboran dapat menutup celah dari optocoupler sehingga optocoupler tidak dapat berfungsi dengan baik.

Ide dasar dari desain sensor kecepatan yang digunakan dalam sistem ini bermula dari ketika meng “on” kan saklar lampu 220 Volt secara keras/memukul saklar, seketika itu filamen dari lampu tersebut menjadi putus.



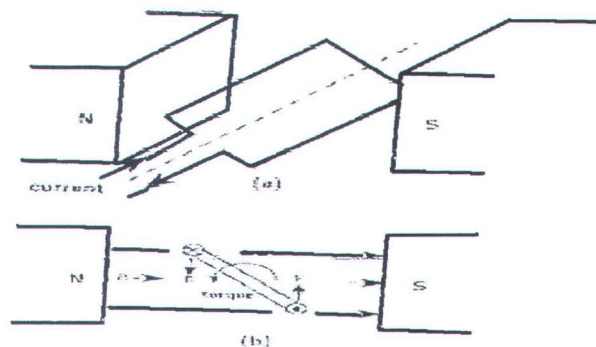
Gambar 3.12. Rangkaian lampu 220 Vac

Gambar 3.12. merupakan gambar instalasi lampu sederhana pada rumah. Kejadian serupa terulang pada IC TTL yang mudah rusak ketika power supply IC tersebut di “on”kan atau di “off “kan. Dengan kejadian tersebut saya membuat rangkaian sederhana seperti gambar 3.13.



Gambar 3.13. Rangkaian Sederhana

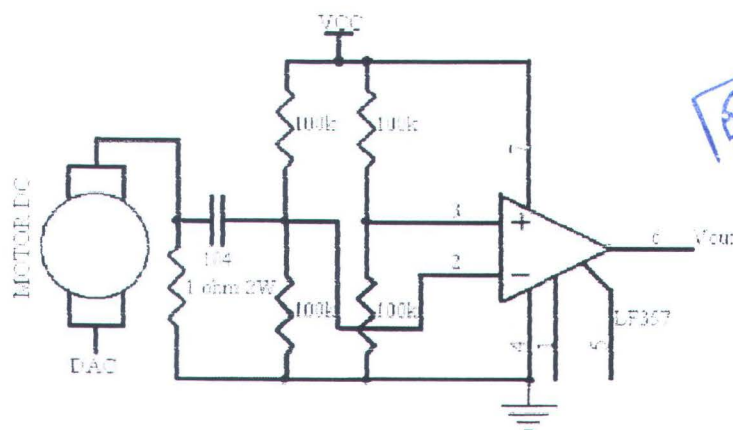
Pada gambar 3.13. terdapat dua buah rangkaian dengan saklar terbuka “on” dan saklar tertutup “off”. Tegangan pada resistor pada saklar terbuka adalah 0 Volt dan tegangan pada resistor pada saklar tertutup adalah 5 volt. Tegangan pada resistor tersebut diukur dengan menggunakan AVO meter digital. Penunjukkan nilai tegangan pada avometer sesaat setelah saklar di tutup “on” bernilai lebih dari 5 Volt dan penunjukkan nilai tegangan sesaat setelah saklar di buka “off” bernilai kurang dari 0 volt. Dapat disimpulkan nilai tegangan akan berada sedikit diatas dan sedikit dibawah ketika terjadi perubahan posisi saklar. Pada gambar 3.14. arah arus bergerak dari kutub + ke resistor dan menuju ke kutub – battery. Battery kutub (+) diasumsikan sebagai medan magnet (N) dan battery kutub (-) sebagai medan magnet (S) sedangkan saklar diasumsikan sebagai brush yang terdapat lilitan. Untuk lebih jelasnya dapat dilihat gambar 3.14.



Gambar 3.14. Cara kerja motor DC

Pada gambar 3.14. posisi brush dapat sejajar ataupun menghalangi arah magnet sehingga dapat diasumsikan seperti saklar yang “on” atau “off”, karena diasumsikan sebagai saklar maka terjadi perbedaan sedikit keatas ataupun kebawah pada tegangan yang diberikan pada kumparan. Jika tegangan sumber 12 volt maka disaat brush berputar tegangan 12 volt tersebut akan bergerak sedikit keatas ataupun kebawah. Ayunan perbedaan tegangan tersebut dapat diamati pada osciloskop.

Besarnya perbedaan tegangan tersebut tergantung pada besarnya sumber tegangan pada motor, semakin besar tegangan sumber semakin besar pula perbedaan tegangan. Besarnya ayunan kurang lebih 50 mV. Semakin cepat putaran motor semakin banyak pula ayunan yang dihasilkan. Banyaknya ayunan tersebut dapat mewakili kecepatan putaran motor. Agar ayunan dapat dihitung maka besarnya ayunan tadi harus stabil (tidak variatif) dan pada level TTL. Tegangan ayunan pada level mili volt. Gambar 3.15. adalah gambar rangkaian comparator.



Gambar 3.15. Rangkaian comparator

Capasitor 10uF berfungsi untuk menghambat tegangan DC, sehingga hanya ayunan yang dihasilkan yang dapat melewati capasitor tersebut. Dengan rangkaian comparator seperti pada gambar 3.15. berapapun perubahan pada motor akan dapat dirubah menjadi level tegangan TTL.

Desain seperti diatas dapat menghemat ruang penempatan rotary encoder, optocoupler dan kabel. Pengujian telah dilakukan ayunan yang dihasilkan sesuai dengan kecepatan pada motor tersebut.

3.2.3 Sensor Jarak

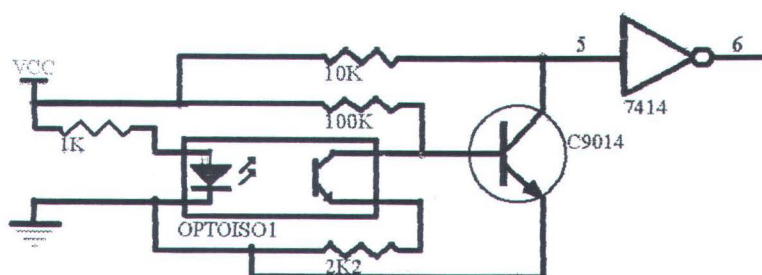
Jarak antara lubang kaki IC berkisar 2.5 mm, jarak antara lubang resistor (axial 0.3) berkisar 7.5 mm. Dilihat dari jarak antara lubang tersebut dapat disimpulkan

motor diharapkan dapat bergerak setiap 0.5 mm. Motor stepper yang digunakan dapat bergerak untuk setiap langkahnya berkisar 0.3 mm. Perhitungan pergerakan dari tiap tiap posisi dapat dihitung melalui berapa banyak pulsa yang diberikan pada motor stepper tersebut. Perhitungan jarak dengan menghitung banyaknya pulsa yang diberikan pada motor stepper terdapat banyak kesalahan terutama disaat motor stepper pertama kali bergerak. Karena pertama kali bergerak motor stepper cenderung slip/pulsa telah diberikan akan tetapi motor tidak bergerak. Setiap akan bergerak terjadi slip, kesalahan slip 0.3 mm, kesalahan yang sedikit untuk pergerakan tersebut. Akan tetapi pada proses mengebor PCB motor sering kali berhenti untuk melakukan pengeboran, maka akumulasi kesalahan slip akan dihitung dan akan bertambah banyak. Perhitungan jarak dengan mengasumsikan banyaknya pulsa yang diberikan tidak dapat mewakili jarak yang dihitung.

Cara yang lain untuk mengukur jarak pergerakan adalah ditempatkannya rotary encoder pada gear/poros motor stepper. Banyaknya pulsa yang dihasilkan rotary encoder tersebut dihitung untuk mengetahui jarak yang telah ditempuh. Permasalahan slip juga akan terjadi, karena poros/gear tersebut telah berputar akan tetapi beban pada sumbu X atau sumbu Y belum bergerak karena pertama kali motor mulai bergerak belt/sabuk akan mengencang sebelum sumbu tersebut bergerak, pada proses ini rotary encoder telah mengirim pulsa untuk dihitung padahal posisi sumbu belum bergerak. Cara yang kedua dan pertama memiliki kesamaan permasalahan.

Agar tidak terjadi kesalahan slip tersebut maka yang diukur harus pada posisi sumbu sehingga pada saat belt atau sabuk mengencang posisi sumbu belum bergerak maka seharusnya sensor tersebut juga masih belum mengirim sinyal. Ide dasar pengukuran jarak pada titik sumbu adalah dengan mengamati rotary encoder. Rotary

encoder adalah lubang lubang yang berputar. Lubang lubang ini dapat juga diartikan dengan gelap (halangan) dan terang (tidak terhalang), opto coupler yang ditempatkan pada rotary encoder tersebut akan terhalang (gelap) dan tidak terhalang (terang). Sehingga akan menghasilkan pulsa. Dengan “gelap dan terang” tersebut dapat dilihat pada ujung pengaris, untuk memudahkan membaca pengaris, terdapat tanda gelap dan terang, setiap 1 cm pengaris terdapat 10 gelap dan 10 terang, setiap gelap berjarak 1 mm dan setiap terang berjarak 1 mm. Pengaris tersebut dapat juga disebut dengan horisontal encoder. Pada titik gelap terdapat tinta sehingga tinta tersebut akan menghalangi cahaya

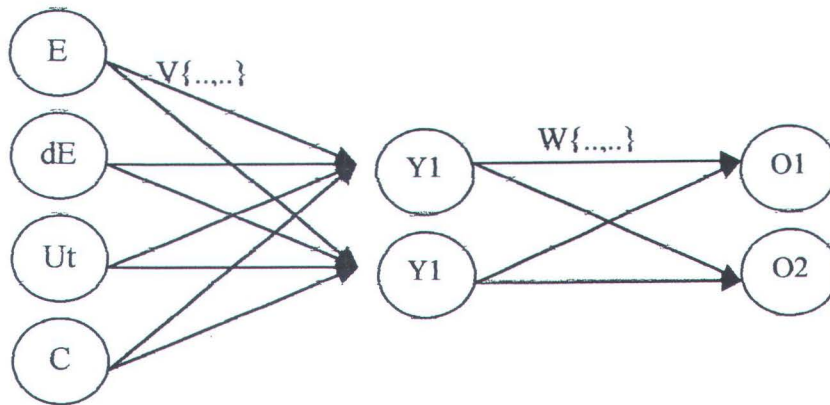


Gambar 3.16. Rangkaian optocoupler

dari opto coupler untuk menuju ke rx. Setelah mengetahui dari horisontal encoder/pengaris, dengan memasang rangkaian optocoupler telah dapat berfungsi sebagai sensor jarak dengan ketelitian 0.5 mm. Pada gambar 3.16. merupakan gambar dari rangkaian optocoupler. Perubahan resistansi dari rx yang akan mengaktifkan basis dari transistor, output dari colector transistor berupa tegangan 0 volt dan 5 volt.

3.3 KONSTRUKSI SOFTWARE

3.3.1 Metoda Neural Network PI controller



Gambar 3.17. Arsitektur Backpro

Fungsi aktivasi yang digunakan pada sistem ini adalah fungsi sigmoid bipolar (range $[-1,+1]$)

$$f(y) = \left[\frac{2}{1 + e^{-y}} \right] - 1 \dots\dots\dots (3.1)$$

Dengan turunannya

$$f'(y) = \frac{1}{2} [1 + f(y)] [1 - f(y)] \dots\dots\dots (3.2)$$

Fungsi aktivasi untuk jaringan propagasi-balik harus memiliki beberapa karakteristik : kontinyu, dapat didiferensiasikan, dan monoton tidak turun. Juga lebih diinginkan demi efisiensi komputasi, turunan fungsinya juga mudah dihitung. Biasanya, fungsinya diharapkan untuk bersaturasi atau mendekati maksimum dan minimumnya secara asimtotik.

Algoritma pembelajaran pada sistem ini terdiri dari beberapa langkah

Langkah 1. Inisialisasi bobot (biasanya diberi nilai acak yang kecil)

Langkah 2. Membandingkan syarat untuk berhenti, jika belum memenuhi melanjutkan langkah selanjutnya.

Langkah 3. Menghitung total sinyal input pada layar 1

$$y(1) = (E * V[1,1]) + (dE * V[1,2]) + (Ut * V[1,3]) + (C * V[1,3]) \dots\dots\dots (3.3)$$

$$y(2) = (E * V[2,1]) + (dE * V[2,2]) + (Ut * V[2,3]) + (C * V[2,3]) \dots\dots\dots (3.4)$$

Langkah 4. Menghitung total sinyal input pada layar 2

$$o(1) = (f(y1) * w[1,1]) + (f(y2) * w[1,2]) \dots\dots\dots (3.5)$$

$$o(2) = (f(y1) * w[2,1]) + (f(y2) * w[2,2]) \dots\dots\dots (3.6)$$

Menghitung total sinyal input pada tiap tiap layar dengan fungsi aktifasinya

$$f(y1) = \left[\frac{2}{1 + e^{-y1}} \right] - 1 \dots\dots\dots (3.7)$$

$$f(y2) = \left[\frac{2}{1 + e^{-y2}} \right] - 1 \dots\dots\dots (3.8)$$

$$f(o1) = \left[\frac{2}{1 + e^{-o1}} \right] - 1 \dots\dots\dots (3.9)$$

$$f(o2) = \left[\frac{2}{1 + e^{-o2}} \right] - 1 \dots\dots\dots (3.10)$$

Dibawah ini adalah cuplikan program untuk langkah 3 dan 4

```

procedure tform3.output_node;
begin
  for j:=1 to maks_J do
  begin
    for i:=1 to maks_I do Y[j]:=Y[j] + (Z[i] * V[j,i]);
    Y[j]:=(2/(1+exp(-(lamda_y*Y[j]))))-1;
  end;

  for k:=1 to maks_K do
  begin
    for j:=1 to maks_J do O[k]:=O[k] + (Y[j] * W[k,j]);
    O[k]:=(2/(1+exp(-(lamda_o*O[k]))))-1;
  end;
end;

```

Langkah 5. Setiap unit output menghitung total sinyal masukan terbobot (menghitung selisih).

$$dO[1] = \frac{1}{2} \sum (D[1] - O[1])^2 \dots\dots\dots (3.11)$$

$$dO[2] = \frac{1}{2} \sum (D[2] - O[2])^2 \dots\dots\dots (3.12)$$

Langkah 6. Layer 1, menghitung total sinyal masukan terbobot (menghitung selisih).

$$dy[1] = \frac{1}{2} (1 - y(1)^2) \sum dO[1] * W \dots\dots\dots (3.13)$$

$$dy[2] = \frac{1}{2} (1 - y(2)^2) \sum dO[2] * W \dots\dots\dots (3.14)$$

Dibawah ini adalah cuplikan program untuk langkah 5 dan 6

```

procedure tform3.hitung_sinyal_error;
var temp:real;
begin
  for k:=1 to maks_K do delta_O[k]:=0.5*(D[k]-O[k])*(1-(O[k]*O[k]));
  for j:=1 to maks_J do
    begin
      temp:=0;
      for k:=1 to maks_K do temp:=temp+(delta_O[k]*W[k,j]);
      delta_y[j]:=0.5*(1-(Y[j]*Y[j]))*temp;
    end;
  end;
end;

```

Langkah 7. Menghitung nilai v dan w atau bobot baru

$$V_{New} = V_{Old} + n_y * dy * input \dots\dots\dots (3.15)$$

$$W_{New} = W_{Old} + n_o * do * Y \dots\dots\dots (3.16)$$

Dibawah ini adalah cuplikan program untuk langkah 7

```

procedure tform3.hitung_bobot_baru;
begin
  for j:=1 to maks_J do
    begin
      for i:=1 to maks_I do
        V[j,i]:=V[j,i]+(n_y*delta_y[j]*Z[i]);
      end;

      for k:=1 to maks_K do
        begin
          for j:=1 to maks_J do
            W[k,j]:=W[k,j]+(n_o*delta_o[k]*Y[j]);
          end;
        end;
      end;
    end;
end;

```

Langkah 8. $f(o1)$ dan $f(o2)$ merupakan nilai Konstanta (K_p dan K_i) pada PI controler

$$U_t = \text{Error} * K_p + \text{Derror} * K_i \dots \dots \dots (3.17)$$

Dibawah ini adalah cuplikan program untuk langkah 8

```
signal:=(error*o[1])+(j_error*o[2]);
```

Langkah 9. Kembali ke langkah 2

Sinyal kontrol (U_t), harus dikonversi menjadi data bit agar dapat merubah data ladder R-2R

Tabel 3.1. konversi U_t ke Data Ladder

U_t	Data Ladder
-1	0 d
-0.5	64 d
0	128 d
0.5	192 d
1	255 d

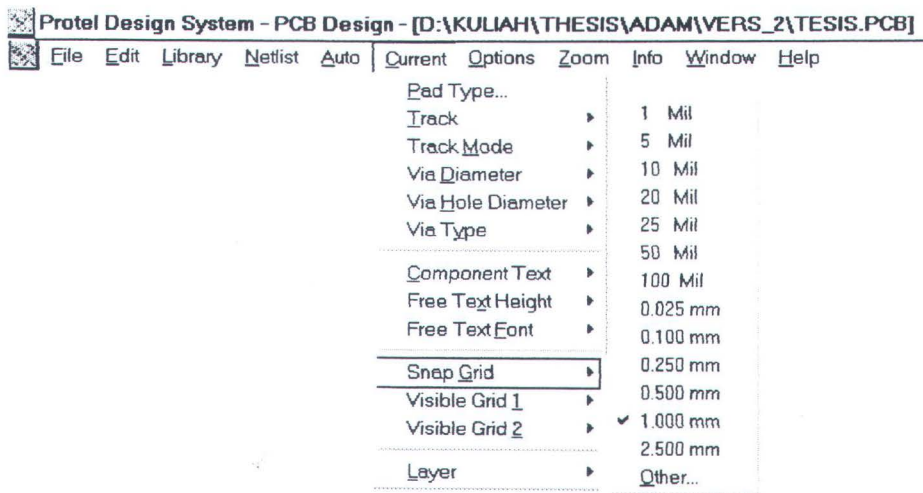
3.3.2 Metoda konversi file PCB

Pada sistem ini menggunakan aplikasi protel design system atau biasa disebut protel for windows. Untuk membuat design PCB dapat menggunakan dua buah cara yaitu melalui schematic (Eda Client) ataupun secara manual, langsung pada protel for windows. Setelah tahap design pada software protel tersebut selesai, ada beberapa hal yang perlu diperhatikan sebelum membuat file.

Hal hal yang perlu diperhatikan adalah

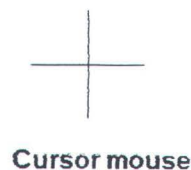
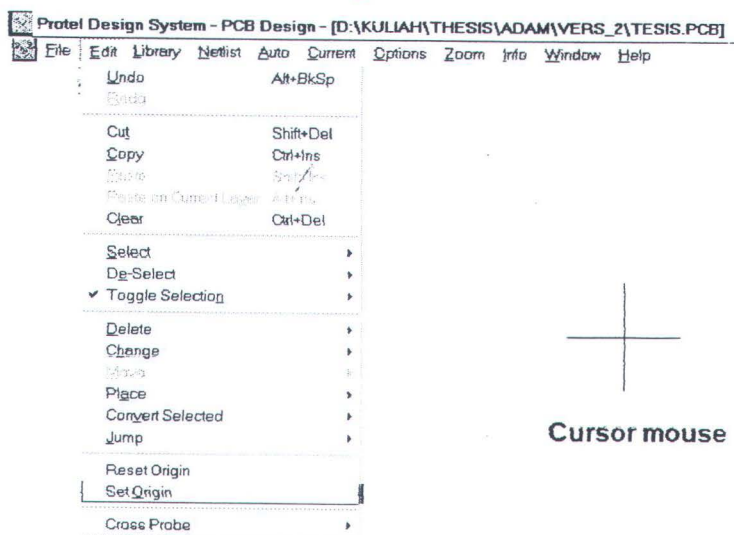
1. Snap grid -> harus diset sesuai dengan resolusi mesin bor tersebut. Pada alat ini resolusinya 1 mm untuk setiap kisi (grid). Posisi snap grid pada menu current lalu

sub menu Snap grid maka keluar pilihan mulai dari 1 mil – 100 mil, 0.025 mm – 2.5 mm. Maka dipilih 1 mm agar sesuai dengan program dari mesin bor tersebut.



Gambar 3.18. Menu snap grid

2. Set Origin -> memberikan titik awal koordinat x,y atau memberikan koordinat 0,0. Pemberian koordinat tersebut nantinya akan menjadi referensi dari mesin bor sebagai posisi awal. Posisi set origin terletak pada menu Edit lalu tekan/click set origin, setelah diclick, kursor mouse akan berubah, lalu kursor tadi harus digerakkan pada koordinat awal yang diinginkan, setelah pada posisi telah ditentukan lalu click ditekan lagi.



Gambar 3.19. Menu Set origin

3. NC Drill -> sub menu NC Drill terdapat pada menu File, yang bertujuan untuk merubah ke bentuk koordinat dalam file text. Cara menjalankan menu NC drill seperti cara menyimpan. Ada beberapa metoda untuk mengetahui maksud dari file tersebut. Dibawah ini merupakan contoh hasil dari menu NC Drill

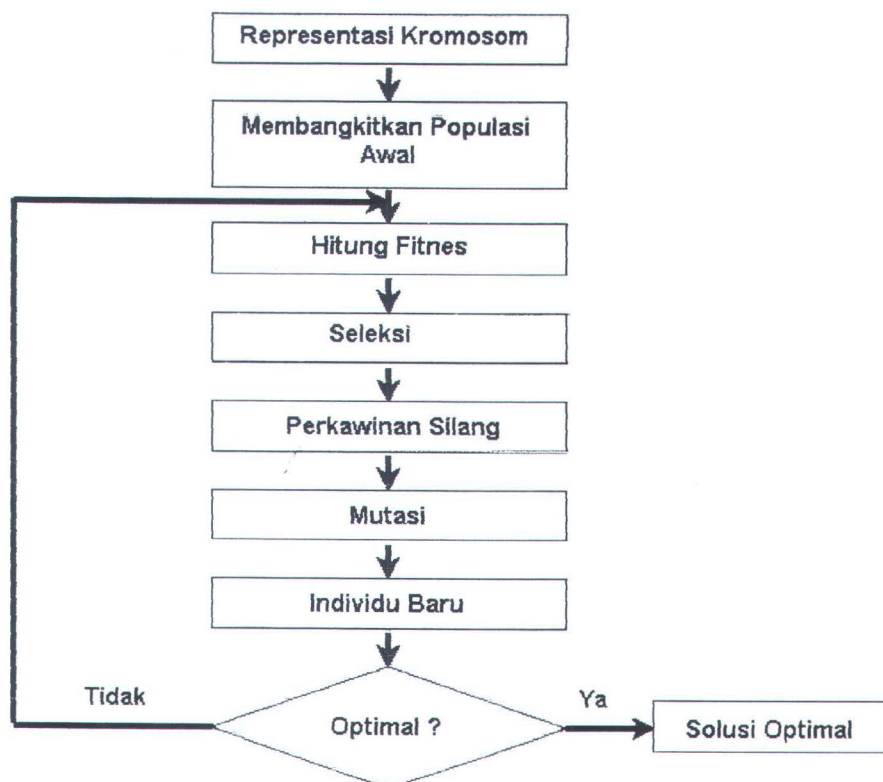
```
1.      M72
2.      M48
3.      T1F00S00C0.711
4.      T2F00S00C0.813
5.      T3F00S00C2.997
6.      %
7.      T01
8.      X0016Y0012
9.      T02
10.     X0005Y000438
11.     Y000692
12.     Y000946
13.     Y0012
14.     X001262
15.     Y000946
16.     Y000692
17.     Y000438
18.     T03
19.     X002Y0008
20.     M30
```

No 1 sampai 20, merupakan tambahan agar memudahkan dalam penulisan, pada file hasil sesungguhnya tidak terdapat nomer tersebut. File hasil konversi ini di sortir/diurutkan berdasarkan kesamaan besarnya lubang. Pada baris ke 3 terdapat "T1F00S00C0.711" T1 yang dimaksud adalah indek untuk lubang pertama, 4 angka dari kiri merupakan besarnya lubang. Dengan demikian pada T1 besarnya adalah 0.711 mm. Pada baris ke 4 terdapat "T2F00S00C0.813" T2 yang dimaksud adalah indek untuk lubang kedua, 4 angka dari kiri merupakan besarnya lubang. Dengan demikian pada T2 besarnya adalah 0.813 mm. Pada baris ke 5 terdapat "T3F00S00C2.997" T4 yang dimaksud adalah indek untuk lubang ketiga, 4 angka dari kiri merupakan besarnya lubang. Dengan demikian pada T3 besarnya adalah 2.977 mm. Baris ke 4 "%" sebagai indikator bahwa indek telah berakhir. Baris ke 7

“T01” merupakan awal dari koordinat untuk lubang 0.711 mm, maka baris ke 8 untuk lubang berukuran 0.711 mm. Baris ke 8 “X0016Y0012” X merupakan koordinat x pada posisi 16 mm dan Y pada posisi 12 mm. Baris ke 9 “T02” merupakan awal dari koordinat untuk lubang 0.813 mm. Baris ke 10 “X0005Y000438” X merupakan koordinat x pada posisi 5 mm dan Y pada posisi 4.38 mm. Lebar character untuk x maksimal 4 digit untuk satuan mm, jika terdapat lebih dari 4 digit maka digit selanjutnya untuk posisi dibelakang koma. Baris ke 11 “Y000692” pada baris ke 11 koordinat yang ada hanya sumbu Y pada posisi 6.92 mm maka sumbu X mengikuti koordinat baris ke 10. Demikian selanjutnya sampai akhir dari file.

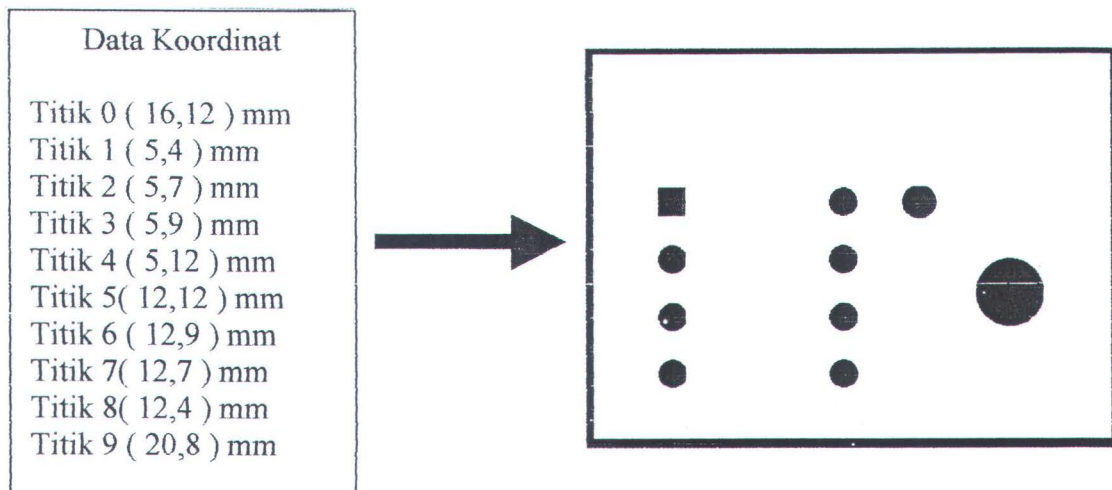
3.3.3 Metoda Genetik Algoritma

Tujuan dari penggunaan metoda genetik algoritma agar jarak lintasan robot lebih efisien sehingga total waktu pengeboran semakin singkat.



Gambar 3.20. Algoritma Genetika

Permasalahan dari mesin bor ini adalah bagaimana mesin bor tepatnya sumbu Z dapat mengebor setiap posisi dengan jarak lintasan yang paling pendek. Sebagai contoh 10 titik koordinat seperti pada contoh diatas.



Titik 1 sampai 10 diasumsikan sebagai nilai integer 0 s/d 9 sebanyak 11 gen, satu gen terakhir merupakan duplikat dari gen pertama.

Contoh01234567891

Yang diinginkan GA ini adalah jarak terpendek sedangkan pada tahap seleksi yang dipertahankan adalah nilai maksimum maka langkah awal menentukan jarak tak terhingga dengan asumsi 1000 cm atau 10.000 mm merupakan jarak lintasan terjauh ataupun tak terhingga, selisih antara jarak tak terhingga dengan jarak lintasan terpendek didapat nilai maksimum. Untuk lebih jelasnya titik A dan Titik B berjarak 1000 mm sedangkan titik B dan C berjarak 2000 mm, maka titik A dan B mempunyai jarak terpendek dibandingkan titik B dan C. Sedangkan fungsi fitnes dari titik A dan B bernilai 99.000 mm dan titik B dan C bernilai 98.000, fungsi fitnes titik A dan B lebih besar dibanding dengan titik B dan C. maka didapat fungsi fitnes dinyatakan dalam rumus adalah

$$Fitness = 1000000 - \sum \sqrt{(X_i - X_{i-1}) + (Y_i - Y_{i-1})} \dots\dots\dots (3.18)$$

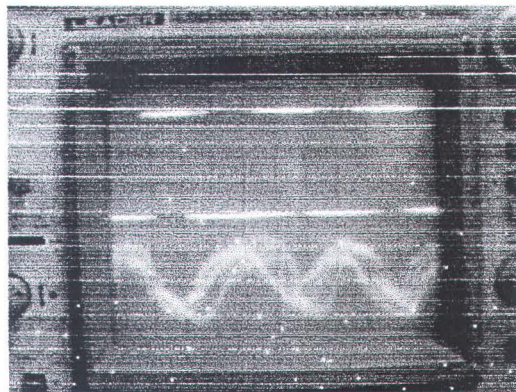
dengan melihat Gambar 3.20. langkah selanjutnya dari algoritma genetika adalah membangkitkan populasi awal, dengan dibangkitkannya 10 individu dengan 11 gen secara acak dengan nilai integer 0-9, langkah selanjutnya adalah mencari nilai fitness terbesar dari setiap individu. Langkah selanjutnya seleksi roulette wheel untuk memilih induk dengan melihat proporsitas dari setiap individu, dimana setiap individu mendapat bagian sesuai dengan nilai fitnessnya. Langkah selanjutnya crossover yaitu perkawinan silang antara 2 posisi awal dan akhir dari gen yang akan dikawinkan secara acak. kemudian kemudian dilakukan penukaran nilai gen induk 1 dan induk 2 dari posisi awal sampai dengan posisi akhir untuk diperoleh anak 1 dan anak 2. Langkah selanjutnya adalah Mutasi dilakukan dengan cara menentukan dua posisi gen secara acak (posisi 0 sampai dengan 9) dan melakukan pertukaran nilai gen pada kedua posisi tersebut, nilai gen ke 1 diduplikasi ke nilai gen ke 9. Langkah selanjutnya adalah membentuk individu baru. Anak hasil perkawinan silang dan mutasi menjadi generasi baru untuk dilakukan proses regenerasi. Pada generasi berikutnya, individu terbaik (nilai fitness terbesar) dapat dipertahankan

BAB IV

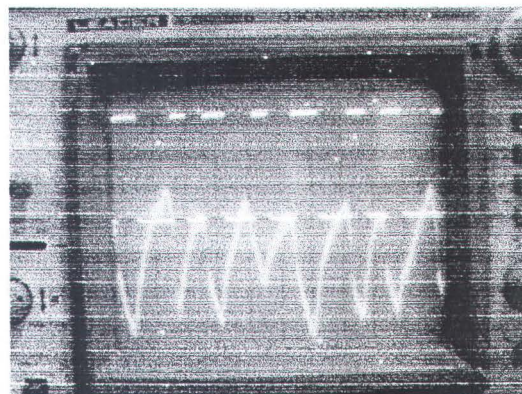
PENGUJIAN SISTEM

4.1 Pengujian Sensor Kecepatan

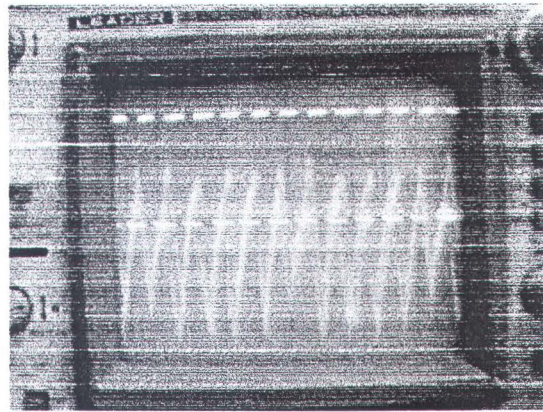
Pengujian sensor kecepatan dengan menggunakan alat bantu Osciloskop. Pengujian pertama chanal Time/div menunjukkan 0.5 uS, pada kanal 1 menunjukkan 1 Volt / div, kanal 2 menunjukkan 50mV/Div. Pada gambar osciloskop terdapat dua buah sinyal, yaitu sinyal kotak dan sinyal sinus. Sinyal sinus merupakan sinyal asli yang dihasilkan oleh motor DC sedangkan sinyal kotak hasil dari sinyal sinus yang telah dikomparator.



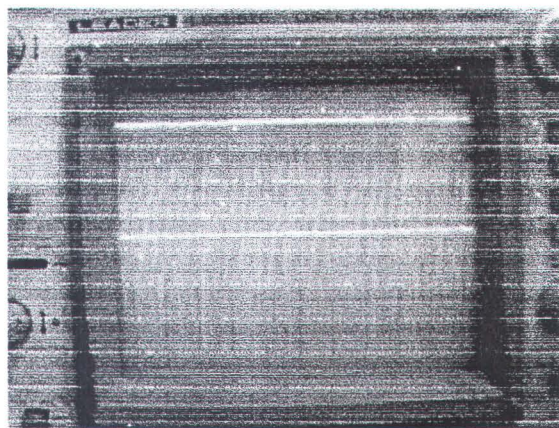
Gambar 4.1 Time/Dive 0.5 uS, Data Ladder R-2R = 32H



Gambar 4.2. Time/Dive 0.5 uS, Data Ladder R-2R = 64H

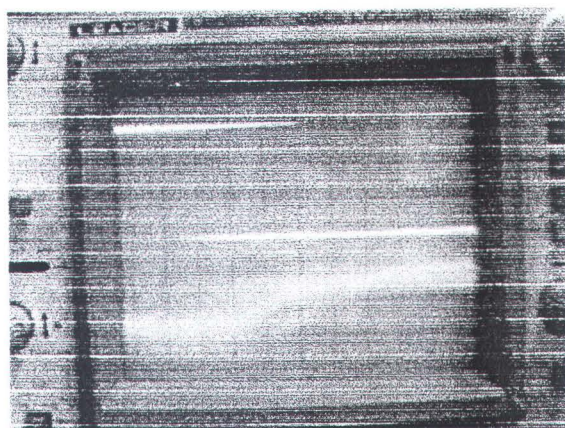


Gambar 4.3. Time/Dive 0.5 uS, Data Ladder R-2R = 96H

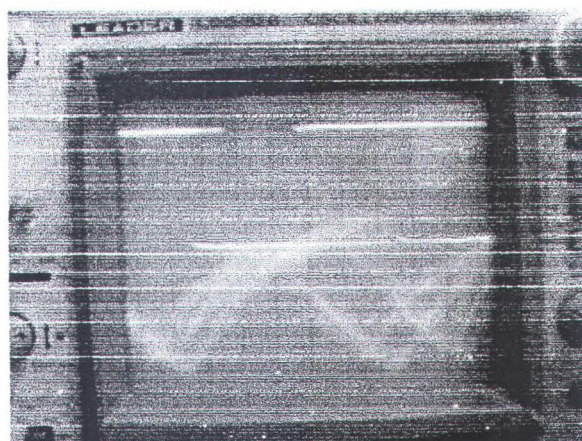


Gambar 4.4. Time/Dive 0.5 uS, Data Ladder R-2R = 0C8H

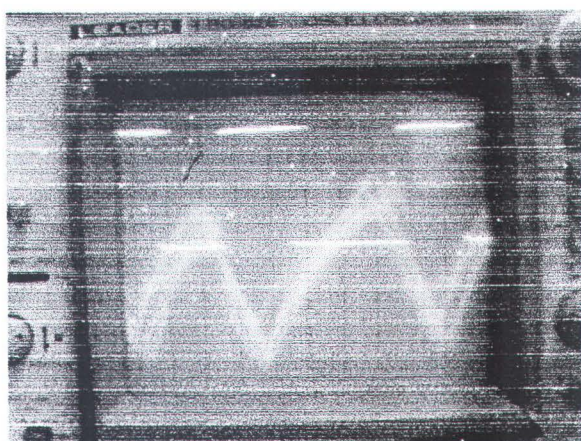
Pengujian kedua kanal Time/div menunjukkan 0.1 uS, pada kanal 1 menunjukkan 1 Volt/div, kanal 2 menunjukkan 50mV/Div. Pada gambar osciloskop terdapat dua buah sinyal, yaitu sinyal kotak dan sinyal sinus. Sinyal sinus merupakan sinyal asli yang dihasilkan oleh motor DC sedangkan sinyal kotak hasil dari sinyal sinus yang telah dikomparator.



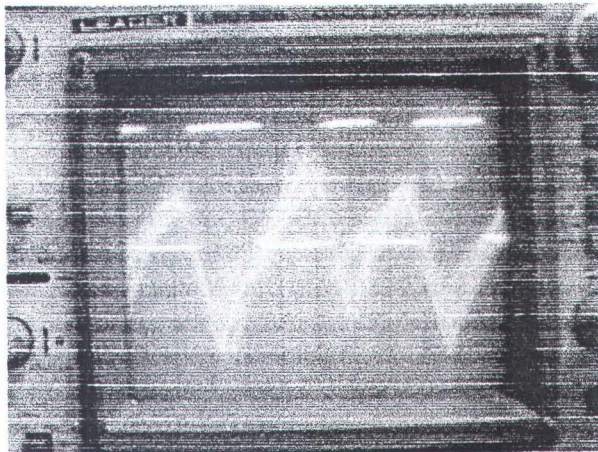
Gambar 4.5. Time/Dive 0.1 uS, Data Ladder R-2R = 032H



Gambar 4.6. Time/Dive 0.1 uS, Data Ladder R-2R = 64H



Gambar 4.7. Time/Dive 0.1 uS, Data Ladder R-2R = 96H



Gambar 4.8. Time/Dive 0.1 uS, Data Ladder R-2R = 0C8H

4.2 Pengujian ladder R-2R

Tabel 4.1 Pengujian *Ladder* R-2R tanpa motor

Input (Data bit)	V out (Volt)	I out (Ampere)
00 D	1.2	0
25 D	2.5	0
50 D	4.5	0
100 D	9.3	0
125 D	11.7	0
150 D	14.1	0
175 D	16.5	0
200 D	18.9	0
225 D	21.3	0
255 D	24.2	0

Input (Data bit) berasal dari komputer yang dikirim melalui serial ke mikrokontroler pada port 2 dan Vout diukur pada transistor dalam keadaan motor DC tidak dihubungkan / dilepas.

4.3 Pengujian sensor posisi

4.3.1 Sumbu X

Pengujian sensor posisi pada sumbu X dilakukan dalam beberapa tahap. Tahap pertama robot menuju posisi 10 Cm atau 100 mm dengan 10 kali pemberhentian. Pengujian dengan metoda ini bertujuan agar dapat menghitung total kesalahan pengukuran yang diakibatkan kesalahan pengukuran sensor tiap 1 mm. Titik awal dari pengukuran dilihat dari posisi sensor optocoupler.

Tabel 4.2. Pengukuran sumbu X dengan ketelitian 10 mm

Input (mm)	Posisi Sensor (Cm)	Pengukuran (mm)	Error (mm)
00	8.9	00	
10	9.9	10	
20	10.9	10	
30	12	11	
40	13	10	
50	14.1	11	
60	15.2	11	
70	16.2	10	
80	17.2	10	
90	18.2	10	
100	19.1	9	
100	$(19.1-8.9) = 10.2$	102	+2

Tabel 4.3. Pengukuran sumbu X dengan ketelitian 20 mm

Input (mm)	Posisi Sensor (Cm)	Pengukuran (mm)	Error (mm)
00	8.9	00	
20	11	21	
40	13.2	22	
60	15.5	23	
80	17.5	20	
100	19.7	22	
100	$(19.7-8.9) = 10.8$	108	+8

Tabel 4.4. Pengukuran sumbu X dengan ketelitian 30 mm

Input (mm)	Posisi Sensor (Cm)	Pengukuran (mm)	Error (mm)
00	8.9	00	
30	12.2	33	
60	15.3	31	
90	18.5	32	
100	19.6	11	
100	$(19.6-8.9) = 107$	107	+6

Tabel 4.5. Pengukuran sumbu X dengan ketelitian 50 mm

Input (mm)	Posisi Sensor (Cm)	Pengukuran (mm)	Error (mm)
00	8.9	00	
50	14.4	55	
100	19.7	53	
100	$(19.7-8.9) = 108$	108	+8

Tabel 4.6. Pengukuran sumbu X dengan ketelitian 100 mm

Input (mm)	Posisi Sensor (Cm)	Pengukuran (mm)	Error (mm)
00	8.9	00	
100	19.5	106	
100	$(19.5-8.9) = 106$	106	+6

Error yang terjadi pada pengukuran sumbu X diakibatkan karena penempatan sensor optocoupler relatif terbuka / terkena cahaya, akibat dari penempatan sensor yang relatif terbuka tersebut adalah resistansi dari sensor sedikit berkurang disaat terkena halangan hitam pada pengaris. Berkurangnya resistansi tersebut karena cahaya luar ikut masuk pada rx pada sensor.

4.3.2 Sumbu Y

Pengujian sensor posisi pada sumbu Y dilakukan dalam beberapa tahap. Pada pengukuran sumbu Y sedikit berbeda dengan sumbu X, pada sumbu Y tidak dicantumkan posisi sensor dikarenakan untuk melihat posisi sensor/pengaris mengalami kesulitan karena optocoupler tepat berada dibawah sumbu X. Pengukuran dilakukan dengan menggunakan pengaris dari posisi terluar.

Tahap pertama robot menuju posisi 10 Cm atau 100 mm dengan 10 kali pemberhentian. Pengujian dengan metoda ini bertujuan agar dapat menghitung total kesalahan pengukuran yang diakibatkan kesalahan pengukuran sensor tiap 1 mm. Titik awal dari pengukuran dilihat dari posisi sensor optocoupler

Tabel 4.7. Pengukuran sumbu Y dengan ketelitian 10 mm

Input (mm)	Pengukuran (mm)	Error (mm)
00	20	
10	30	
20	40	
30	50	
40	60	
50	70	
60	80	
70	90	
80	100	
90	110	
100	120	
100	$(120-20) = 100$	0

Tabel 4.8. Pengukuran sumbu Y dengan ketelitian 20 mm

Input (mm)	Pengukuran (mm)	Error (mm)
00	21	
20	41	
40	61	
60	81	
80	111	
100	121	
100	$(121-21) = 100$	0

Tabel 4.9. Pengukuran dengan ketelitian 30 mm

Input (mm)	Pengukuran (mm)	Error (mm)
00	21	
30	51	
60	81	
90	110	
100	121	
100	$(121-21) = 100$	0

Tabel 4.10. Pengukuran sumbu Y dengan ketelitian 50 mm

Input (mm)	Pengukuran (mm)	Error (mm)
00	21	
50	71	
100	121	
100	$(121-21) = 100$	0

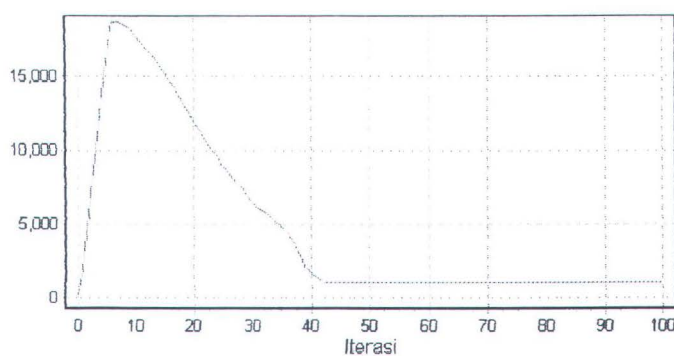
Tabel 4.11. Pengukuran sumbu Y dengan ketelitian 100 mm

Input (mm)	Pengukuran (mm)	Error (mm)
00	21	
100	121	
100	$(121-21) = 100$	0

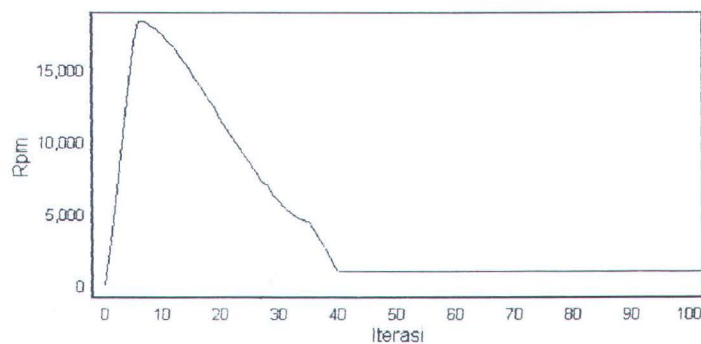
Pada sumbu Y tidak terdapat error pengukuran karena sensor optocoupler terlindungi dari cahaya langsung . cahaya langsung tersebut terhalang oleh sumbu X

4.4 Pengujian NNPT

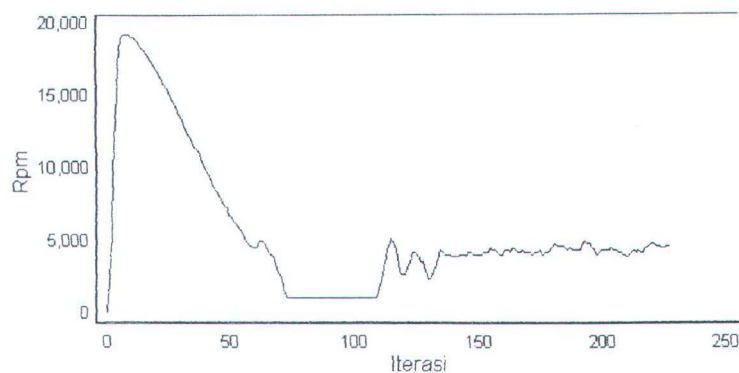
Pengujian NNPT bertujuan untuk mengetahui secara langsung respon dari sistem jika di hubungkan langsung dengan motor DC. Hasil dari pengujian ini berupa tampilan dari software. Parameter $K_p = 0.46$ dan $K_i = 0.2$, parameter tersebut dibuat tetap akan tetapi setpoint (Rpm) di rubah mulai dari 1000 Rpm – 30.000 Rpm



Gambar 4.9. Percobaan 1 dgn setpoint 1000 Rpm

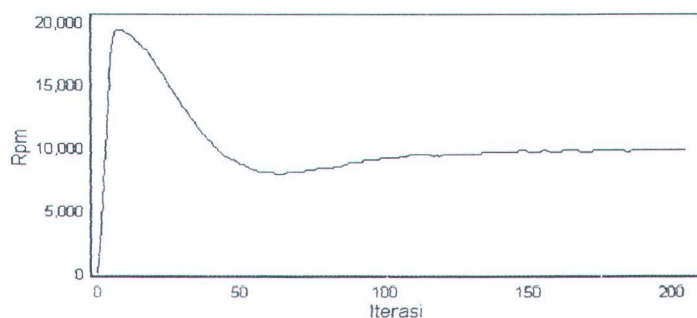


Gambar 4.10. Percobaan 2 dgn setpoint 1000 Rpm

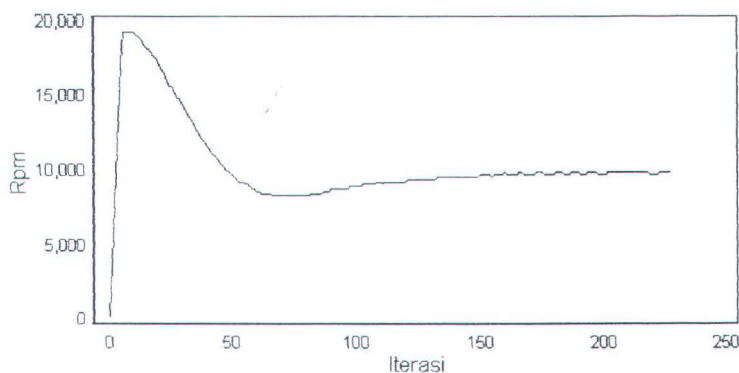


Gambar 4.11.. Percobaan 1 dgn setpoint 5000 Rpm

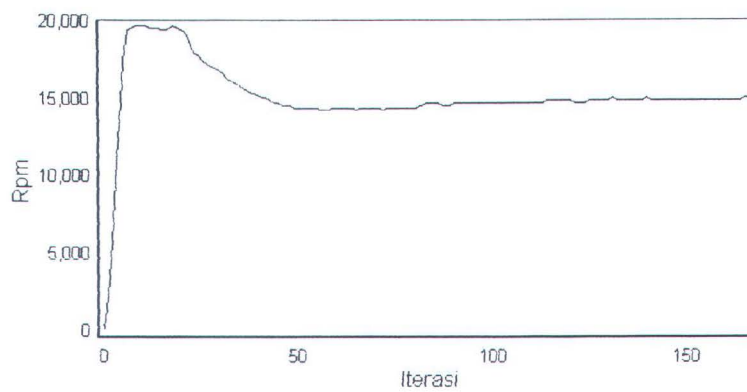
Pada gambar 4.10. dan gambar 4.11. terjadi *overshoot* yang besar dan untuk menuju kestabilan relatif lama serta kecepatan motor relatif tidak stabil. Pada rpm 1000 dan 5000 tidak dapat digunakan untuk proses mengebor karena kecepatan motor tidak stabil.



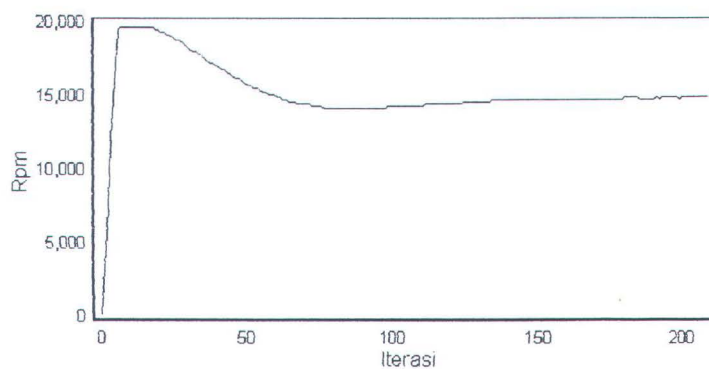
Gambar 4.12. Percobaan 1 dgn setpoint 10.000 Rpm



Gambar 4.13. Percobaan 2 dgn setpoint 10.000 Rpm

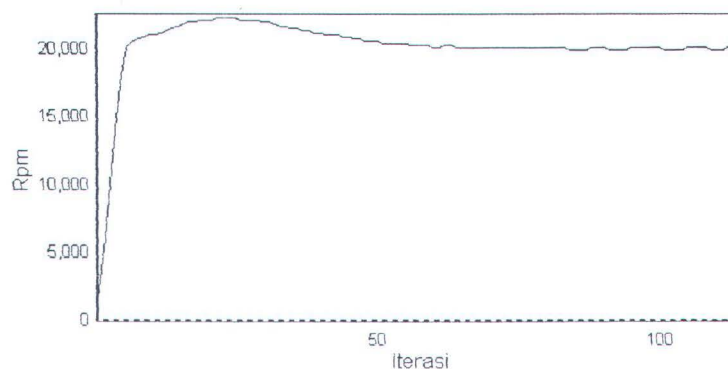


Gambar 4.14. Percobaan 1 dgn setpoint 15.000 Rpm

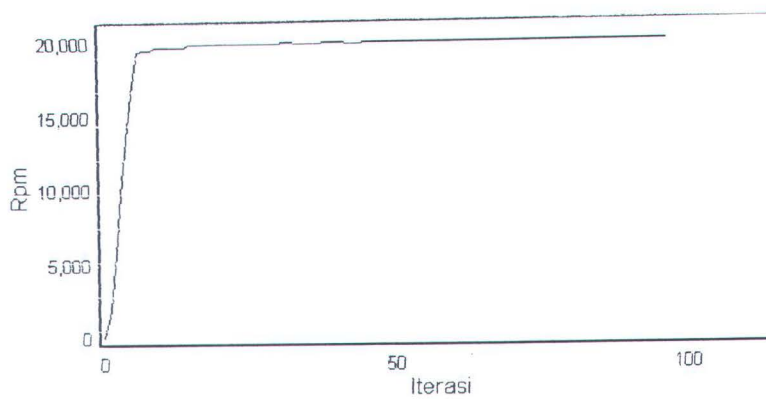


Gambar 4.15. Percobaan 2 dgn setpoint 15.000 Rpm

Pada gambar 4.12. dan gambar 4.15. terjadi *overshoot* yang besar dan untuk menuju kestabilan relatif lama ,pada percobaan ini kecepatan motor DC relatif stabil. Pada rpm 10000 dan 15000 belum dapat digunakan untuk proses mengebor karena masih terjadi error yang besar dan membutuhkan waktu yang lama untuk menuju pada titik kecepatan yang stabil.

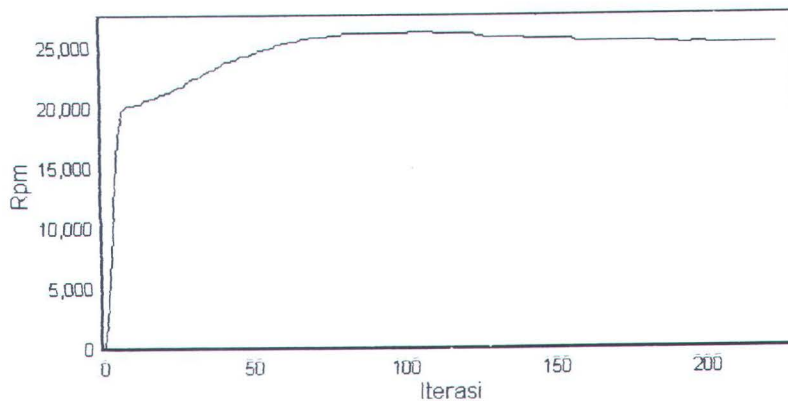


Gambar 4.16. Percobaan 1 dgn setpoint 20.000 Rpm

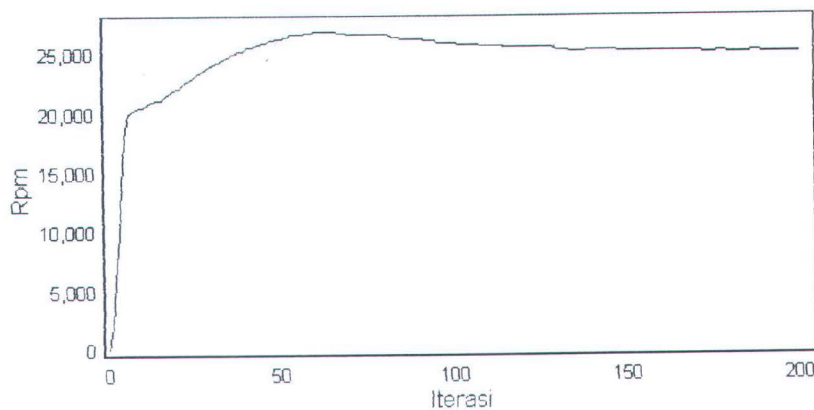


Gambar 4.17. Percobaan 2 dgn setpoint 20.000 Rpm

Pada gambar 4.16. dan 4.17 dilakukan percobaan dengan setpoint 20.000 rpm pada percobaan ini tidak terjadi *overshoot* dan hanya memerlukan waktu yang singkat untuk menuju ke titik kecepatan yang stabil. Pada setpoint 20.000 rpm telah dapat digunakan untuk proses mengebor.

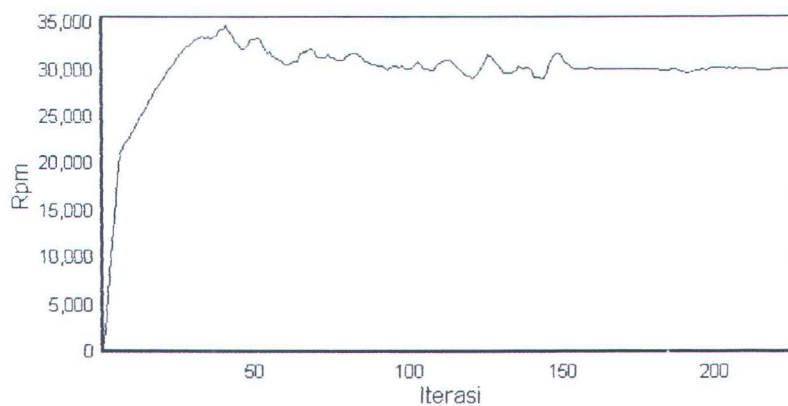


Gambar 4.18. Percobaan 1 dgn setpoint 25.000 Rpm

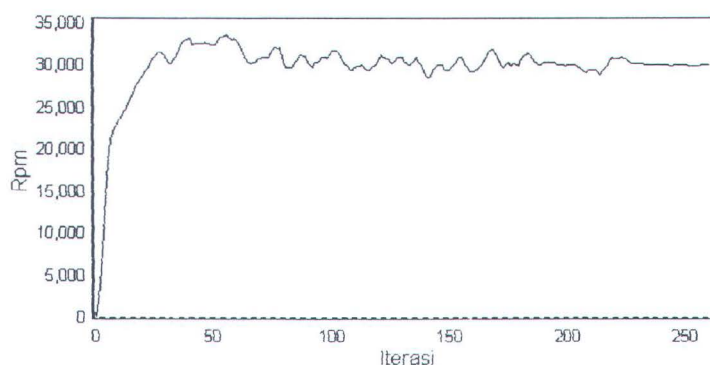


Gambar 4.19. Percobaan 2 dgn setpoint 25.000 Rpm

Pada gambar 4.18 dan 4.19 dengan setpoint 25.000 rpm tidak terjadi *overshoot* akan tetapi membutuhkan waktu yang lama untuk menuju ketitik kecepatan yang stabil. Pada kecepatan ini tidak dapat digunakan untuk proses mengebor.



Gambar 4.20. Percobaan 1 dgn setpoint 30.000 Rpm

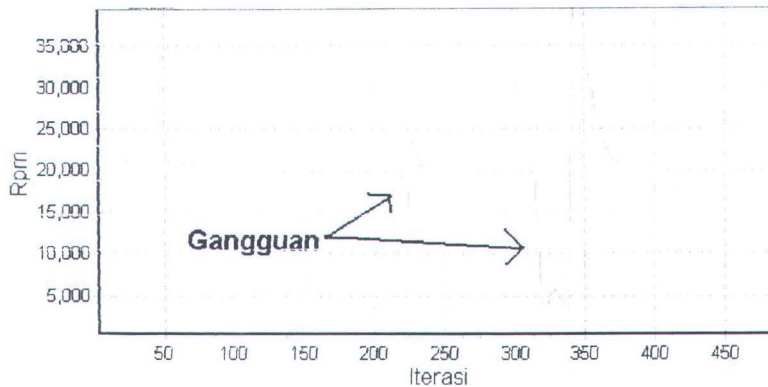


Gambar 4.21. Percobaan 2 dgn setpoint 30.000 Rpm

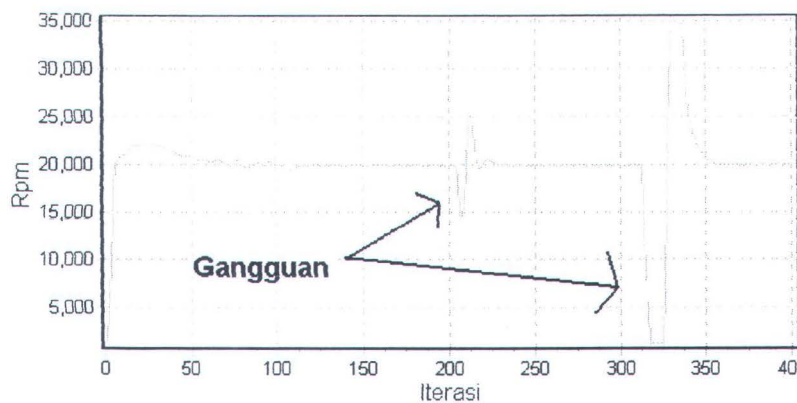
Pada gambar 4.20 dan 4.21 dengan setpoint 30.000 rpm relatif kecepatannya tidak stabil. Pada percobaan ini tidak dapat digunakan untuk prose mengebor

Setelah melakukan pengujian NNPT terhadap motor DC dapat diamati pada kecepatan 20.000 Rpm motor DC relatif lebih optimal, mulai dari respon menuju ke kestabilan (*stady state*) yang cepat dan error puncak (*Overshoot*) yang tidak begitu besar. Maka diputuskan pada kecepatan 20.000 rpm tersebut mesin bor akan berkerja. Pada pengujian dibawah ini bor diset pada kecepatan 20.000 Rpm , di saat

kecepatan bor telah stabil , pada ujung motor DC atau tepatnya pada mata bor di beri gangguan.



Gambar 4.22. Percobaan 1 dgn setpoint 20.000 Rpm dgn gangguan



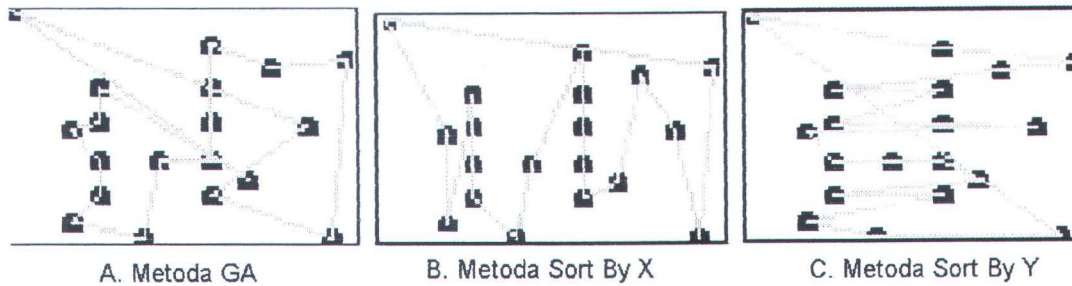
Gambar 4.23. Percobaan 2 dgn setpoint 20.000 Rpm dgn gangguan

Pada gambar 4.23 dan 4.24. gangguan diberikan berupa pemberian beban secara tiba – tiba yang mengakibatkan putaran motor DC terganggu. Percobaan ini bertujuan untuk mengetahui respon dari NNPT jika terjadi gangguan secara mendadak.

4.5 Pengujian Algoritma Genetika

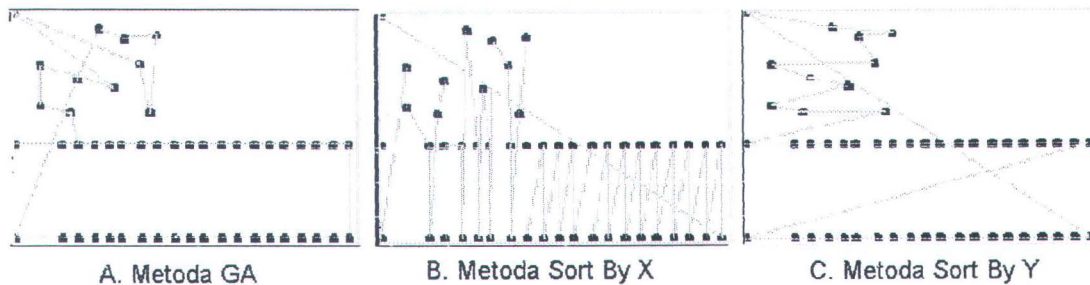
Agar dapat mengetahui bahwa metoda genetik algoritma telah berkerja dengan baik, maka pengujian metoda GA harus dibandingkan dengan metoda konvensional seperti metoda Sort by X yang mana metoda ini akan melakukan pengurutan

berdasarkan kesamaan koordinat X dan sort by Y yang mana metoda ini akan mengurutkan berdasarkan kesamaan koordinat Y.



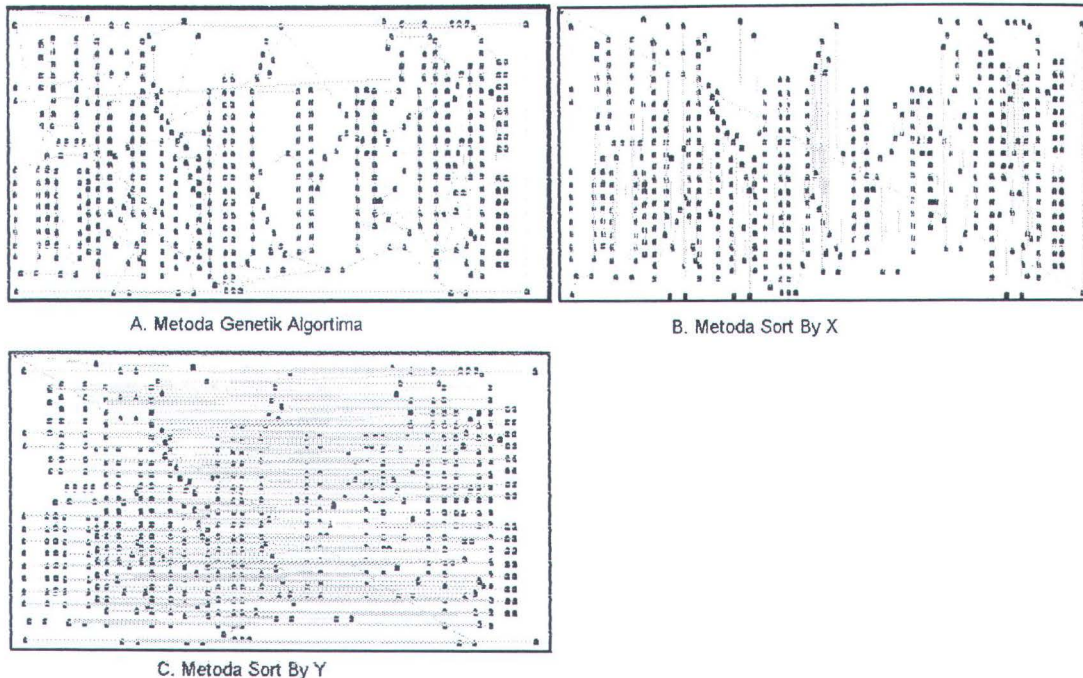
Gambar 4.24. Hasil dari tiga buah metoda (1)

Pada gambar 4.24. total lubang adalah 18, dengan menggunakan metoda GA lintasan yang ditempuh sejauh 13 mm, lintasan yang ditempuh dengan metoda sort by X sejauh 12 mm dan lintasan yang ditempuh dengan menggunakan metoda sort by Y sejauh 17 mm.



Gambar 4.25. Hasil dari tiga buah metoda (2)

Pada gambar 4.25. total lubang adalah 50, dengan menggunakan metoda GA lintasan yang ditempuh sejauh 27 mm, lintasan yang ditempuh dengan metoda sort by X sejauh 86 mm dan lintasan yang ditempuh dengan menggunakan metoda sort by Y sejauh 34 mm.



Gambar 4.26. Hasil dari tiga buah metoda (3)

Pada gambar 4.26. total lubang adalah 541, dengan menggunakan metoda GA lintasan yang ditempuh sejauh 333 mm, lintasan yang ditempuh dengan metoda sort by \bar{X} sejauh 470 mm dan lintasan yang ditempuh dengan menggunakan metoda sort by Y sejauh 2249 mm.

Semakin banyak lubang yang akan dibor akan mempengaruhi dari kinerja metoda algoritma genetik. Pada lintasan yang kompleks metoda GA dapat mencapai efisiensi lebih dari 50 persen dibandingkan dengan metoda konvensional.

Tabel 4.12 Data percobaan dengan Jumlah lubang 0 s/d 49

Jumlah Lubang	Jarak Tempuh Metoda GA (Cm)	Jarak Tempuh Metoda Sort by X (Cm)	Jarak Tempuh Metoda Sort by Y (Cm)
3	1	1	1
4	2	2	2
8	4	5	7
8	6	10	8
24	15	19	27
24	14	24	19
16	12	14	8
18	11	11	14
32	16	23	35
32	19	32	24
32	19	32	24
32	18	25	35
40	21	45	29
40	21	32	48
40	13	21	66
48	23	25	76
48	18	80	26
48	31	33	65
48	26	62	37
49	27	86	34
Jarak Tempuh Rata - rata	15.85	29.1	29.25

Pada tabel 4.12 merupakan hasil uji coba terhadap jumlah lubang antara 0 s/d 49 dengan melakukan uji coba sebanyak 20 kali, dapat disimpulkan

- Efisiensi metoda GA terhadap metoda *sort by X* = 45.53%
- Efisiensi metoda GA terhadap metoda *sort by Y* = 45.81%

Tabel 4.13 Data percobaan dengan Jumlah lubang 50 s/d 132

Jumlah Lubang	Jarak Tempuh Metoda GA (Cm)	Jarak Tempuh Metoda Sort by X (Cm)	Jarak Tempuh Metoda Sort by Y (Cm)
50	28	78	51
72	40	88	65
72	42	88	65
80	34	102	59
80	40	91	98
80	36	141	50
87	27	47	48
88	24	46	46
88	42	67	135
88	39	137	63
88	37	61	132
96	41	149	58
92	32	55	62
124	44	136	71
124	50	96	107
124	48	91	97
132	54	78	139
132	48	167	77
132	62	90	162
132	50	110	101
Jarak Tempuh Rata - rata	40.9	95.9	84.3

Pada tabel 4.13 merupakan hasil uji coba terhadap jumlah lubang antara 50 s/d 132 dengan melakukan uji coba sebanyak 20 kali, dapat disimpulkan

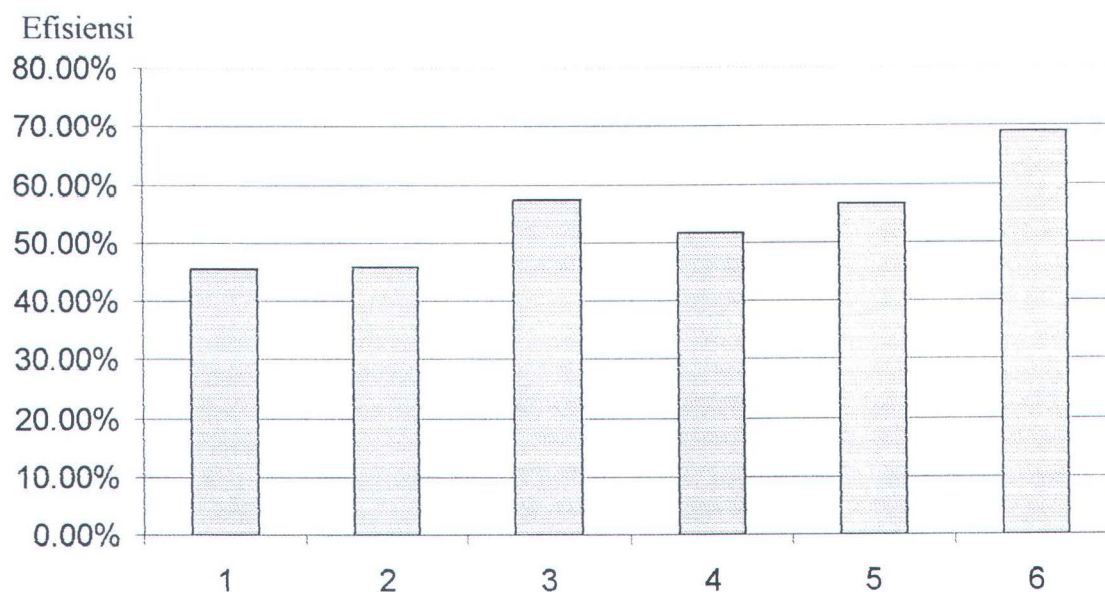
- Efisiensi metoda GA terhadap metoda *sort by* X = 57.35%
- Efisiensi metoda GA terhadap metoda *sort by* Y = 51.48%

Tabel 4.14 Data percobaan dengan Jumlah lubang 133 s/d 741

Jumlah Lubang	Jarak Tempuh Metoda GA (Cm)	Jarak Tempuh Metoda Sort by X (Cm)	Jarak Tempuh Metoda Sort by Y (Cm)
174	103	144	321
174	143	220	178
174	105	140	337
182	95	312	135
361	202	359	644
369	237	357	872
377	238	388	973
377	230	524	691
377	235	685	652
384	209	350	637
385	208	398	651
389	209	335	809
421	244	622	688
429	257	351	1089
429	221	480	790
429	239	485	758
665	426	2076	759
665	482	751	2195
673	400	750	2334
741	640	2073	992
Jarak Tempuh Rata - rata	256.15	590	825.25

Pada tabel 4.14 merupakan hasil uji coba terhadap jumlah lubang antara 133 s/d 741 dengan melakukan uji coba sebanyak 20 kali, dapat disimpulkan

- Efisiensi metoda GA terhadap metoda *sort by X* = 56.58%
- Efisiensi metoda GA terhadap metoda *sort by Y* = 68.96%



Gambar 4.27. Grafik efisiensi metoda GA

Keterangan

1. Efisiensi metoda GA terhadap metoda *sort by* X dengan jumlah lubang antara 0 s/d 49 = 45.53%
2. Efisiensi metoda GA terhadap metoda *sort by* Y dengan jumlah lubang antara 0 s/d 49 = 45.81%
3. Efisiensi metoda GA terhadap metoda *sort by* X dengan jumlah lubang antara 50 s/d 132 = 57.35%
4. Efisiensi metoda GA terhadap metoda *sort by* Y dengan jumlah lubang antara 50 s/d 132 = 51.48%
5. Efisiensi metoda GA terhadap metoda *sort by* X dengan jumlah lubang antara 133 s/d 741 = 56.58%
6. Efisiensi metoda GA terhadap metoda *sort by* Y dengan jumlah lubang antara 133 s/d 741 = 68.96%

BAB V

KESIMPULAN DAN SARAN

5.1 Kesimpulan

Pada tesis ini akan dibangun mekanik robot untuk aplikasi pengeboran PCB. Agar robot dapat melakukan pengeboran diperlukan tiga buah mekanik, yaitu untuk koordinat sumbu X, sumbu Y dan sumbu Z. Pada konstruksi mekanik ini didesain seringan mungkin agar robot dapat bergerak cepat. Untuk membuat mekanik menggunakan sebagian komponen mekanik dari *printer dot matrix double folio*, komponen mekanik printer yang digunakan antara lain pangkon/dudukan, *belt*(sabuk), gear dan motor stepper. Hanya dengan menambah sedikit tambahan komponen dari luar seperti alumunium telah dapat dibangun mekanik robot untuk keperluan bor PCB.

Agar robot mengetahui posisi dimana tiap – tiap sumbu maka diperlukan sensor posisi untuk mengetahuinya. Sensor ini diharuskan dapat mengukur dengan ketelitian 0.1 Cm atau 1 mm karena untuk aplikasi pengeboran ini memerlukan ketelitian yang akurat. Agar dapat mengetahui posisi sumbu secara akurat maka diperlukan dua buah optocoupler. Pada optocoupler pertama berfungsi sebagai referensi/titik nol dan optocoupler kedua berfungsi untuk menghitung pulsa, setiap pulsanya dapat diasumsikan dengan 1 mm. Hasil pengujian sensor posisi pada sumbu X masih terdapat error perhitungan dan hasil pengujian sensor posisi pada sumbu Y tidak terdapat error.

Setelah dapat membangun mekanik robot dan sensor posisi pada robot maka langkah selanjutnya adalah membuat perencanaan pergerakan robot. Jumlah lubang

pada PCB tergantung dari desain mulai dari 0 - 49 lubang, 50 s/d 132 dan 133 s/d 741. Mengingat begitu banyak lubang yang harus dilalui maka perencanaan lintasan yang optimal dapat membuat pergerakan robot menjadi efisien. Metoda untuk perencanaan lintasan pergerakan robot ini menggunakan metoda genetik algoritma, metoda ini cukup optimal dibandingkan dengan metoda konvensional seperti *sort by X* dan *sort by Y*.

- Efisiensi metoda GA terhadap metoda *sort by X* dengan jumlah lubang antara 0 s/d 49 = 45.53%
- Efisiensi metoda GA terhadap metoda *sort by Y* dengan jumlah lubang antara 0 s/d 49 = 45.81%
- Efisiensi metoda GA terhadap metoda *sort by X* dengan jumlah lubang antara 50 s/d 132 = 57.35%
- Efisiensi metoda GA terhadap metoda *sort by Y* dengan jumlah lubang antara 50 s/d 132 = 51.48%
- Efisiensi metoda GA terhadap metoda *sort by X* dengan jumlah lubang antara 133 s/d 741 = 56.58%
- Efisiensi metoda GA terhadap metoda *sort by Y* dengan jumlah lubang antara 133 s/d 741 = 68.96%

Kualitas lubang hasil pengeboran tergantung pada kecepatan putaran motor dan kestabilan putaran motor. Metoda yang digunakan agar dapat menjaga kestabilan kecepatan putaran mata bor adalah NNPT (Neural Network Parameter Tune). Metoda ini merupakan penggabungan antara metoda neural network dan PI. Pada metoda PI diperlukan pengaturan Kp (Konstanta proporsional) dan KI (konstanta Integral). Yang mana pengaturan Kp dan Ki tersebut akan dilakukan oleh neural

network. Hasil pengujian NNPT terhadap motor DC dapat diamati pada kecepatan 20.000 Rpm motor DC relatif lebih optimal, mulai dari respon menuju ke kestabilan (*stady state*) yang cepat dan error puncak (*Overshoot*) yang tidak begitu besar. Maka diputuskan pada kecepatan 20.000 rpm tersebut mesin bor akan berkerja

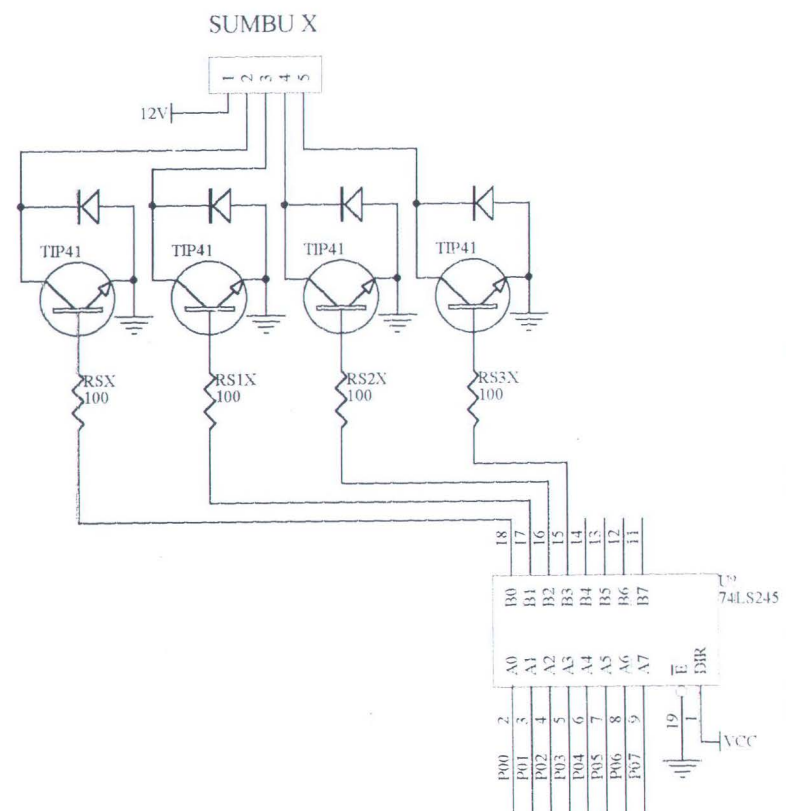
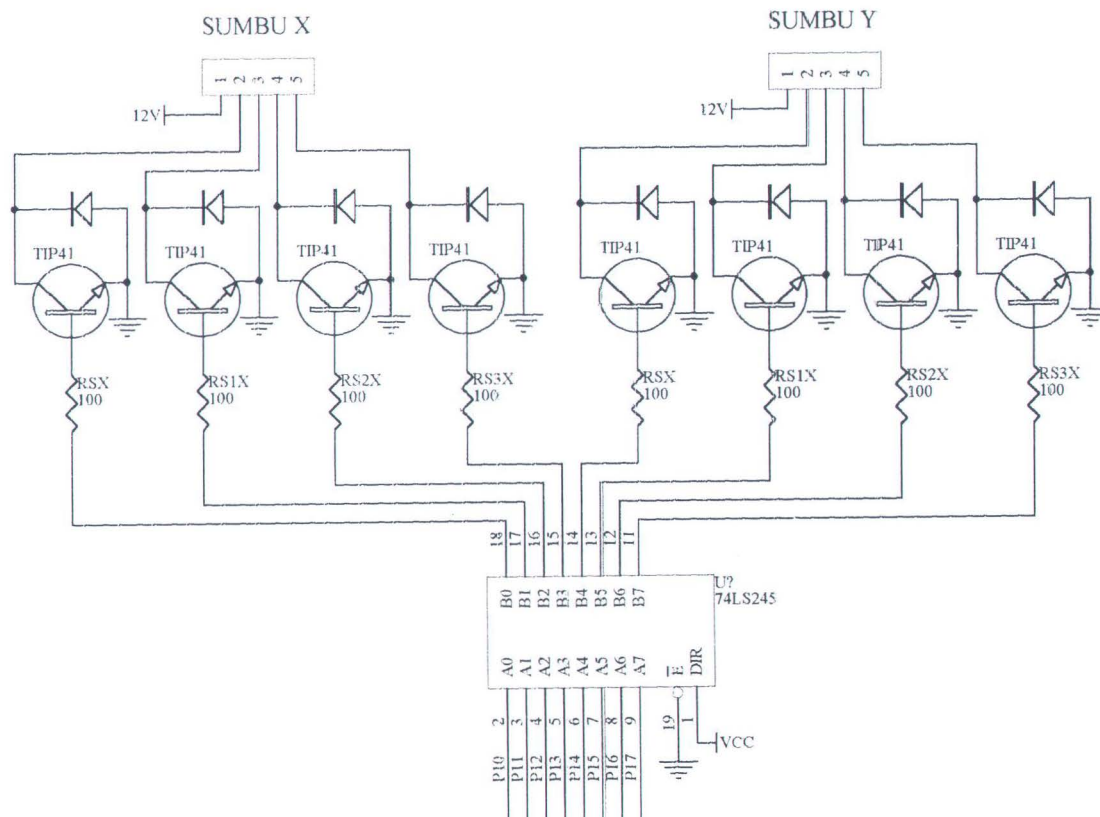
5.2 Saran

Permasalahan yang terjadi pada mekanik adalah antara sumbu X dan sumbu Y. Pada kedua sumbu tersebut sering bergeser/bergerak sehingga pada proses pengeboran terjadi pengeseran posisi yang disebabkan oleh mekanik. Diharapkan untuk perencanaan mekanik agar dibuat kokoh sehingga tidak dapat bergeser antara sumbu X dan sumbu Y. Pergerakan pada mekanik sumbu Z memerlukan waktu yang lama karena pada mekanik ini menggunakan ulir sehingga untuk menggerakkan sumbu Z diperlukan banyak putaran agar sumbu tersebut bergerak. Banyaknya putaran tersebut menjadikan waktu pengeboran relatif lama. Diharapkan untuk pada sumbu z ini dapat didesain tidak menggunakan ulir karena penggunaan ulir relatif memerlukan waktu yang lama untuk pergerakannya.

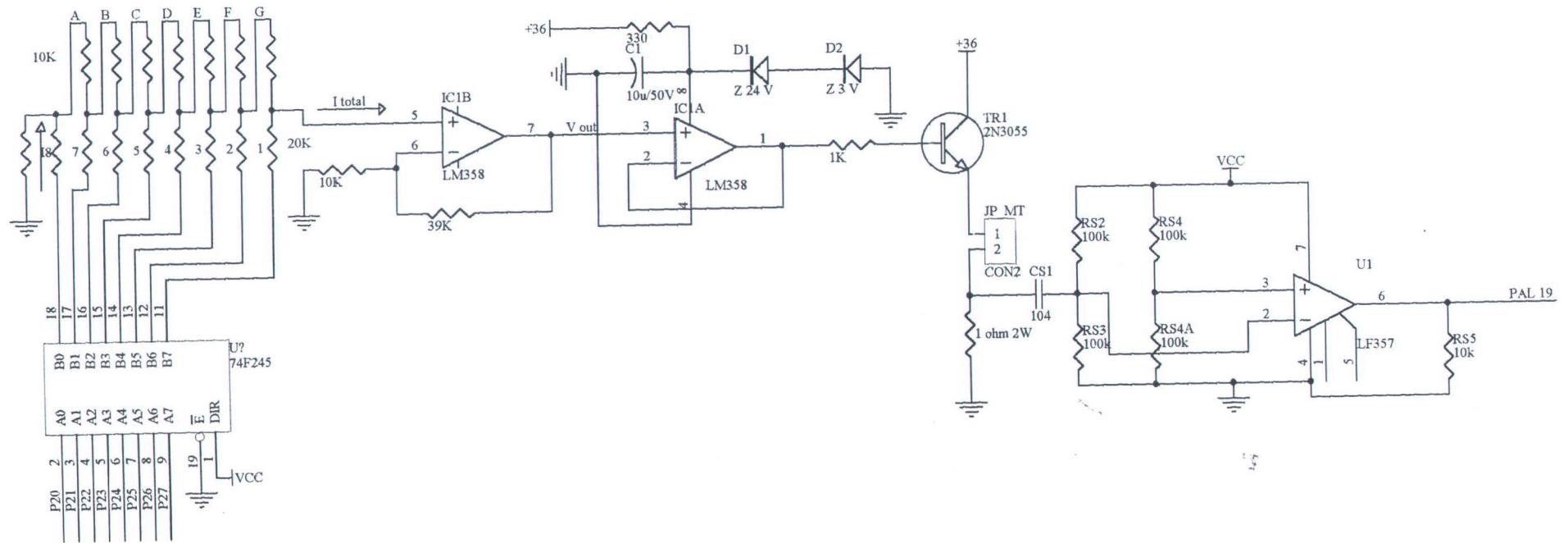
Pengukuran posisi pada sumbu X masih terdapat error, hal ini disebabkan karena penempatan posisi sensor ada diatas yang mengakibatkan cahaya luar dapat secara langsung mempengaruhi kinerja sensor sensor. Penempatan sensor posisi harus terlindung dari cahaya luar yang dapat mengganggu kinerja sensor posisi.

DAFTAR PUSTAKA

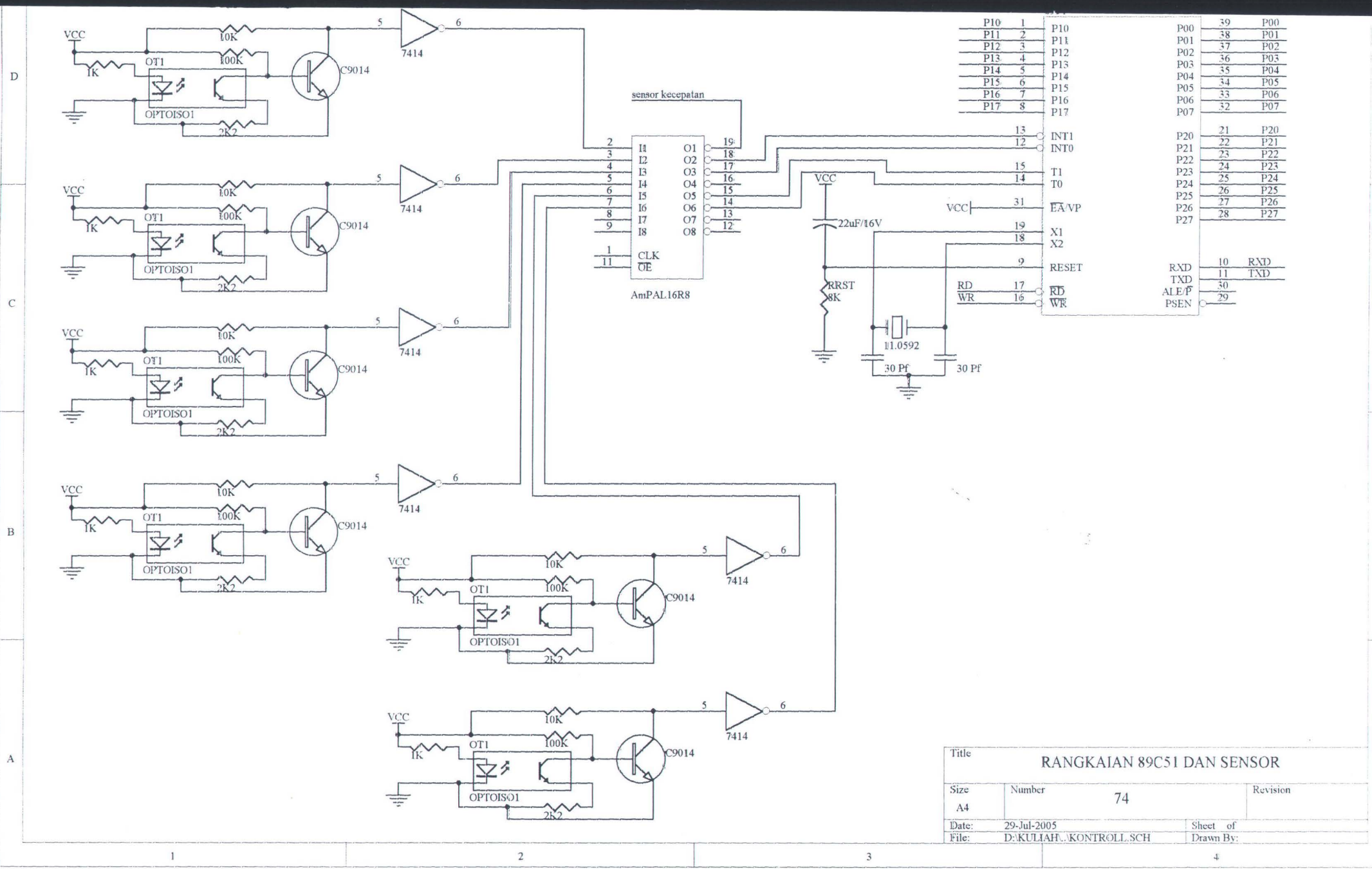
- [1] Zbigniew Michalewicz. (1999), '*Genetic Algorithms + Data Structures = Evolution Programs*', Third Edition. Springer-Verlag Berlin Heidelberg.
- [2] Arabas, J., Michalewicz, Z., and Mulawka, J. GAVaPS. '*AGenetic Algorithm with Varying Population Size*', In [275], pp 73-78.
- [3] Cooper, L., and Steinberg, D. (1970) '*Introduction to Methods of Optimization*', W.B Saunders, London,.
- [4] McGraw – Hill, ROBOTICS: '*Control , Sensing, Vision and Intelligence*', Purdue University.
- [5] Linda Shapiro, '*Computer Vision, Department Of Computer Science*', Michigan State University
- [6] Robert M , '*COMPUTER AND ROBOT VISION I*', University of Washington
- [7] Robert M , '*COMPUTER AND ROBOT VISION II*', University of Washington
- [8] Jacek M, '*Introduction to Neural Systems* ', Info Access Distribution. Ltd.
- [9] Charles A, (1993) '*Modern Industrial Electronics* ', McGraw-Hill.



Title				RANGKAIAN MOTOR STEPPER			
Size		Number		Revision			
A4		72					
Date:		29-Jul-2005		Sheet of			
File:		D:\KULIAH\...STEPPER.SCH		Drawn By:			



Title			RANGKAIAN LADDER R-2R & SENSOR KECAPATAN	
Size	Number	73		Revision
A4				
Date:	29-Jul-2005	Sheet of		
File:	D:\KULIAH\DAC.SCH	Drawn By:		



Title			RANGKAIAN 89C51 DAN SENSOR	
Size	Number	74		Revision
A4				
Date:	29-Jul-2005	Sheet of		
File:	D:\KULIAH\KONTROLL.SCH	Drawn By:		